



Aalto-yliopisto
Insinööritieteiden
korkeakoulu

Jarno Lampainen

IFC-standardin avulla tapahtuvan tiedonsiirron sujuvoittaminen tieto- ja laskentamallien välillä betonirakennesuunnittelussa

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi
diplomi-insinöörin tutkintoa varten.

Espoossa 23.11.2017

Valvoja: Professori Jari Puttonen

Ohjaaja: Diplomi-insinööri Mari Heino

Tekijä Jarno Lampainen

Työn nimi IFC-standardin avulla tapahtuvan tiedonsiirron sujuvoittaminen tieto- ja laskentamallien välillä betonirakennesuunnittelussa

Koulutusohjelma Rakenne- ja rakennustuotantotekniikka

Pää-/sivuaine Rakennetekniikka

Koodi R3001

Työn valvoja Professori Jari Puttonen

Työn ohjaaja(t) DI Mari Heino

Päivämäärä 23.11.2017

Sivumäärä 41

Kieli Suomi

Tiivistelmä

Tieto- ja laskentamallitekniologiat ovat kehittyneet tasaisesti viimeisen 20 vuoden aikana. Samalla on lisääntynyt niiden käyttö rakennesuunnittelun työvälineenä. Seurauksena on ollut kasvava tarve siirtää tietoa tieto- ja laskentamalliohjelmien välillä. Toistaiseksi ei ole kehittynyt yhtä hallitsevaa tapaa tai menetelmää, jolla tiedonsiirto voitaisiin ongelmitta toteuttaa. Yksi potentiaalisimmista vaihtoehdoista ongelman ratkaisemiseksi on IFC-standardi. Sen tarkoituksena on olla neutraali ja ohjelmistoriippumaton tiedonsiirtoformaatti. Tässä diplomityössä testattiin IFC-muotoisen tiedon siirtymistä tieto- ja laskentamallien välillä yksittäisten, muodoltaan monimuotoisten betonirakenneosien avulla. Testien perusteella poikkileikkaukseltaan yksinkertaisten rakenneosien geometria voidaan IFC:n avulla siirtää tietomallista laskentamalliin. Monimutkaisempien, kuten poikkileikkaukseltaan muuttuvien rakenneosien geometrian siirtymisessä on ongelmia. Lisäksi materiaali- ja kuormitustietoja ei saatu testeissä siirtymään. Testitulosten perusteella IFC-muotoinen tiedonsiirto sopii heikosti suurien ja monimuotoisten rakennekokonaisuuksien siirtämiseen tieto- ja laskentamallien välillä. Tutkimuksessa kävi myös ilmi, että käyttäjällä on hyvin vähän keinoja vaikuttaa tiedonsiirron onnistumiseen.

Avainsanat BIM, IFC, laskentamalli, tietomalli, yhteentoimivuus, tekla structures, robot, tiedonsiirto



Author Jarno Lampainen

Title of thesis Generating more flexible IFC based data transfer between building information model and calculation model in design of concrete structures

Degree programme Structural engineering and building technology

Major/minor Structural engineering

Code R3001

Thesis supervisor Professor Jari Puttonen

Thesis advisor M.Sc. (tech) Mari Heino

Date 23.11.2017

Number of pages 41

Language Finnish

Abstract

Since the beginning of this century, building information model (BIM) and structural analysis technologies have steadily improved. It has increased their implementation in structural engineering. Consequence of this development has been greater need for reliable means to transfer data between BIM and structural analysis software. Thus far, there has not emerged a single superior method to execute flawless interoperability between different programs. One possible and potential solution to this problem is IFC standard, which is a platform neutral and open file format specification. In this thesis, IFC based data transfer was tested with individual concrete components. The study shows, that it is possible to transfer concrete components with simple cross-sections via IFC-file between BIM and structural analysis software. However, transferring more complex structures, such as components with invariable cross-sections, have significant problems. Additionally, the transfer of material and load information was not successful. In conclusion, IFC based data transfer between BIM and structural analysis software is not suitable for large and geometrically diverse structures. During the study, it also became evident that users have very few means to affect the success of the data transfer.

Keywords BIM, IFC, building information model, interoperability, tekla structures, robot, data transfer

Alkusanat

Tämä diplomityö on tehty Vahanen Oy:n toimeksiantona kehittämään yhtiön osaamista tieto- ja laskentamallien välisessä tiedonsiirrossa. Työn valvojana toimi rakenne- ja rakennustuotantotekniikan laitoksen professori Jari Puttonen ja ohjaajana Vahanen Oy:n puolelta Mari Heino.

Haluan erityisesti kiittää professori Puttosta kunnan arvoisista kommenteista. Lisäksi kiitän Vahanen Oy:tä kiinnostavasta aiheesta sekä Mari Heinoa pitkämielisyydestä.

Espoo 23.11.2017

Jarno Lampainen

Sisällysluettelo

Tiivistelmä	
Abstract	
Alkusanat	
Sisällysluettelo	1
1 Johdanto	2
2 Tietomalli rakennesuunnittelun työvälineenä	4
2.1 Termien ja sanojen tausta	4
2.2 Rakennuksen tietomalli eri näkökulmista määriteltynä	4
2.2.1 Teknologinen määritelmä ja visio tulevasta	4
2.2.2 Tietomalli oliotietokantana ja sen näennäinen älykkyys	6
2.3 Hyödyt ja edut rakennesuunnittelussa	7
2.4 Tietomallin mallinnusprosessi	9
3 Laskentamallin hyödyntäminen tietomallintavassa rakennesuunnittelussa	11
3.1 Laskentamallin matemaattinen ymmärtäminen ja perusteet	11
3.2 Simulointipohjainen rakennesuunnittelu	14
3.3 Laskentamallin luonti ja analysointi	16
3.4 Yhteentoimivuus ja tiedonsiirtoprosessit	19
3.4.1 Yhteentoimivuuden käsite ja sen tasot	19
3.4.2 Tiedonsiirto yhteisen standardin avulla	21
3.4.3 IFC-standardin visio, haasteet ja mahdollisuudet	23
4 IFC-standardin avulla tapahtuvan tiedonsiirron testaus	24
4.1 Aiempia tutkimuksia ja tutkimusmenetelmän valinta	24
4.2 Tutkimusasetelma, tavoitteet ja oletukset	26
4.3 Geometriatiedon siirtäminen IFC-tiedostoksi	27
4.3.1 Ohjelmien asetukset	27
4.3.2 Geometriatiedon siirtyminen kuvin esitettynä	28
5 Pohdinta	33
5.1 Havainnot tuloksista ja tutkimusprosessista	33
5.2 IFC-pohjaisen tiedonsiirron hyödyntäminen	34
5.3 Tiedonsiirto tieto- ja laskentamallien välillä nyt ja tulevaisuudessa	35
5.4 Tutkielman kriittinen tarkastelu, haasteet ja jatkotutkimus	36
Lähtelueluettelo	38

1 Johdanto

Parin viime vuosikymmenen ajan tietomallintaminen on ollut rakennussuunnittelussa yleinen tutkimuksen kohde. Sitä on vuoroin ylistetty käänteentekevästä ja suunnittelua mullistavana teknologiana, vuoroin taas moitittu työtaakan lisäämisestä ja kankeista ohjelmistoista. Rakennuksen tietomallin näkyvin osuus on kolmiulotteisista rakenneosista muodostettu 3D-malli, mutta tietomallin todellinen hyötypotentiaali on sen mahdollisuudessa liittää rakenneseen erilaista tietoa. Teoriassa tietoa voidaan tallentaa tietomalliin käytettäväksi koko rakennuksen elinkaaren ajan. Tässä tutkielmassa kuitenkin keskitytään tietomallin hyödyntämiseen rakennesuunnittelussa rakennuksen suunnitteluvaiheessa.

Perinteisessä dokumenttipohjaisessa rakennussuunnittelussa ihmiset tuottavat tietoa toisten ihmisten tulkittavaksi, ja tietotekniikka nähdään lähinnä suunnittelun apuvälineenä sekä suunnitelmien esittämisen ja varastoinnin välineenä. Tietomallipohjainen suunnittelu perustuu vielä nykyäänkin pitkälti näihin samoihin perinteisiin lähtökohtiin, mutta tulevaisuuden visiossa se nähdään älykkäänä rakennuksen koko elinkaarta palvelevana teknologiana. Ideologisesti tulevaisuuden tietomallissa kysymys on myös tiedon tuottamisesta tietokoneen ja ohjelmien – ei pelkästään ihmisten tulkittavaksi.

Tietomallitekniikan kanssa samaan aikaan ovat kehittyneet myös erilaiset rakenteiden laskenta- ja analyysiohjelmat. Ohjelmien kirjo on laaja ja ne ovat tyypillisesti pitkälle omaan aihepiiriinsä erikoistuneita. Niillä voidaan mallintaa muun muassa rakenteiden rakennusfysikaalista toimintaa tai tuulen käyttäytymistä rakennusten julkisivujen pinnoilla. Rakennesuunnittelun kannalta merkittävimpiä ovat erilaiset laskentaohjelmat, joiden avulla rakenteiden kuormien ja voimien jakaantumista voidaan mallintaa rakenneosittain tai suurempina kokonaisuuksina.

Tietomalli- ja laskentaohjelmien tietoteknisestä kehityksestä on seurannut loogisesti se, että ohjelmien tuottamaa tietoa pyritään hyödyntämään ja siirtämään eri ohjelmien välillä. Tämä ei kuitenkaan ole ollut ongelmatonta, vaan tiedonsiirrossa on havaittu puutteita ja jopa virheitä. Ongelman ytimenä on, ettei ole pystytty kehittämään aukotonta ja kaiken kattavaa tiedonsiirtomenetelmää tietomallimuotoiselle tiedolle, joka ohjelmasta ja ohjelmistovalmistajasta riippumatta kykenisi siirtämään tietoa tarkasti ja käyttäjän haluamalla tavalla tietomalli- ja laskentaohjelmien välillä. Myöskään toistaiseksi mikään yksittäinen ohjelmistovalmistaja ei ole kyennyt suvereenisti yhdistämään tietomalli- ja laskentaohjelmia yhdeksi ohjelmaksi, jolloin tarve tiedon siirtämiselle eri ohjelmien välillä poistuisi kokonaan.

Tiedonsiirtoon eri ohjelmien välillä tarvitaan aina yhteinen rajapinta, jossa määritetään, miten tiedonsiirto tapahtuu. Rajapinta voidaan toteuttaa monella tapaa, joista yksi on yhteinen standardi. Tietomallimuotoisen tiedon siirtoon on olemassa avoimeen formaattiin perustuva IFC-standardi. Sitä kehittää kansainvälinen ja epäkaupallinen järjestö buildingSMART. Järjestön ja sen kehittämän standardin tavoitteena on, että tietomallimuotoista tietoa voidaan siirtää ohjelmistoriippumattomasti paitsi tietojärjestelmien, myös eri ohjelmistojen välillä. Rakennussuunnittelun kannalta merkittävää on, että IFC:n avulla ei voida siirtää piirustusmuotoista tietoa, vaan se keskittyy niin sanotun oliotiedon eli käytännössä 3D-geometrian ja parametrien siirtämiseen.

Vaikka IFC-standardi ei toistaiseksi ole saavuttanut suvereenia asemaa tiedonsiirtomenetelmänä, on se epäkaupallisuutensa ja avoimuutensa ansiosta yksi potentiaalisimmista keinoista ratkaista tiedonsiirron haasteet tietomallimuotoisen tiedon siirrossa. Eräs sen vahvuuksista on, että monet ohjelmistovalmistajat tukevat IFC-muotoista tiedonsiirtoa omien sisäisten tiedostomuotojen rinnalla. Tosiasia kuitenkin on, että myöskään IFC-pohjainen tiedonsiirto ei ole ongelmatonta, vaan esimerkiksi geometrialtaan monimutkaisten rakenteiden siirtymisessä ohjelmien välillä on havaittu virheitä. Rakennusalan tai rakennesuunnittelun perinteisenä toimenkuvana ei ole ollut ratkaista näitä ongelmia, vaan vastuu ratkaisusta nähdään tyypillisesti tieto- ja ohjelmistotekniikan tehtävänä. Kuitenkin myös rakennusosalalla on hyvä tunnistaa tiedonsiirron mahdollisuudet ja rajoitteet.

Tämän diplomityön tavoitteena on tutkia missä laajuudessa IFC-tiedonsiirtoa voidaan hyödyntää tieto- ja laskentamallien välisessä tiedonsiirrossa betonirakennesuunnittelussa. Lisäksi etsitään rakennesuunnittelijan käytössä olevia käytännön keinoja tiedonsiirron sujuvoittamiseksi. Tyypillisesti tietomalliohjelmissa mallintaminen on helpompaa ja työkalut ovat parempia kuin laskentamalliohjelmissa. Näin ollen tarkastelun pääpaino on siinä, miten tiedonsiirto tietomallista laskentamalliin onnistuu, ja kuinka kattavasti ja tehokkaasti siirrettyä tietoa voidaan laskentamallissa hyödyntää.

Aiemmissa tutkimuksissa on monipuolisesti käsitelty tietomallimuotoisen tiedon siirtämistä rakennussuunnitteluun osallistuvien osapuolten käyttämien tietomalliohjelmien välillä. Lisäksi tutkimuksissa on usein keskitytty vertailemaan eri tiedonsiirtotapoja keskenään ja tarkastelemaan näiden vahvuuksia ja heikkouksia. IFC-standardiin perustuva tiedonsiirto on vain yksi monista eri tavoista siirtää tietoa, joka kuitenkin tämän diplomityön puitteissa halutaan nostaa paremmin esiin. Vähemmälle huomiolle tutkimuksissa on myös jäänyt rakennesuunnittelulle merkittävä tiedonsiirto tieto- ja laskentamallien välillä, joka tässä tutkielmassa nostetaan päärooliin.

2 Tietomalli rakennesuunnittelun työvälineenä

2.1 Termien ja sanojen tausta

Sanalla tietomalli tarkoitetaan tyypillisesti ohjelmistotekniikassa käytettyä tiedon esitystapaa, jossa tieto jäsennetään toisiinsa loogisesti liittyviksi kokonaisuuksiksi. Tietomalleissa keskitytään usein esittämään yhteneviä asiakokonaisuuksia, kuten tämän tutkielman aiheena oleva rakennuksen tietomalli. Se on suora käännös englannin kielen sanoista *building information model*, jonka etukirjaimista syntyvää akronyymiä BIM käytetään myös suomenkielessä. Rakennuksen tietomallista käytetään myös etenkin vanhemmissa kirjoituksissa nimityksiä rakennuksen tuotetietomalli tai sen lyhyempää muotoa tuotemalli.

Sanaa tieto käytetään suomenkielessä varsin väljästi ilmaisemaan lähes kaikkea mahdollista tietoa. Hietanen (2005) kuitenkin erottelee ihmisen ja tietokoneen käsittelemän informaation siten, että tietokone käsittelee dataa, joka vasta ihmisen käsittelemänä muuttuu tiedoksi. Rajanveto perustuu tiedon ymmärtämiseen: tietokone ainoastaan käsittelee tietoa, mutta ei kykene tulkitsemaan ja ymmärtämään sitä. Englannin kielessä nämä kaksi asiaa on erotettu toisistaan siten, että tietokoneen käyttämästä bittimuotoisesta tiedosta (ykkösistä ja nolista) käytetään nimitystä *data*, kun taas ihmisen tulkitsemasta informaatiosta sanaa *knowledge*. Tässä diplomityössä käytetään vierasperäistä lainasanaa data, kun halutaan korostaa tiedon olevan nimenomaan tietokoneen käsittelemää.

Tätä eroa tietokoneen käsittelemän datan ja ihmisen ymmärtämän tiedon välillä, on pyritty poistamaan tietotekniikassa jo pitkään. Asia nousee tosinaan esiin myös rakennuksen tietomallintamisen yhteydessä (esim. Succar 2009), kun pohditaan mihin suuntaan tietomallintaminen on kehittymässä. Nykytilanteessa tietomallimuotoista tietoa lähinnä tallennetaan ja varastoidaan erilaisiin tiedostoihin ja tietojärjestelmiin ihmisten tulkittavaksi. Kuitenkin tulevaisuuden visiona on, että tietomalliohjelmat kykenisivät itse tuottamaan tietoa ja tekemään siitä tulkintoja ihmisten tapaan.

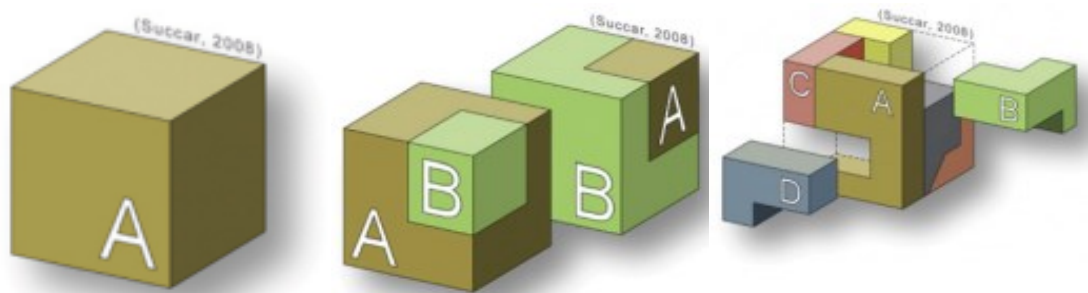
2.2 Rakennuksen tietomalli eri näkökulmista määriteltynä

2.2.1 Teknologinen määritelmä ja visio tulevasta

Rakennuksen tietomallille käytännönläheisen määrittelyn antavat Mäki et al. (2012), jotka näkevät sen teknologiana, joka käsittelee rakennuksen kolmiulotteisia malleja eli 3D-malleja, joihin voidaan sisällyttää rakennusosia koskevaa tietoa. He painottavat määrittelyssään tietoteknistä näkökulmaa, jossa tietomalli käsitetään rakennushankkeeseen osallistuvien toimijoiden yhteisenä tietokantana, jonka kautta suunnittelutietoa voidaan jakaa ja välittää hankkeen osapuolten kesken. Tämän lähestymistavan kannalta tietomallintamisen ideaalitavoitteena on, että kaikki rakennusta koskeva tieto tallennetaan tietomalliin, jossa se on käytettävissä koko rakennuksen elinkaaren ajan suunnittelusta ylläpi-
toon ja aina rakennuksen purkuun asti.

Hietanen (2005) lähestyy tietomallintamista teoreettisemmin. Hänen mielestään tietomallintaminen on paitsi tapa välittää tietoa, myös kokonaan uudenlainen tapa tuottaa ja käsitellä tietoa rakennussuunnittelussa. Tasopiirustusten sijaan rakennuksen suunnitelmat esitetään 3D-malleina, ja rakennelaskelmien ohella suunnittelua ohjataan tietokoneavusteilla analyysillä sekä simulaatioilla. Hietasen (2005) näkemyksessä keskeiseen rooliin nousee ohjelmistotekniikka, joka mahdollistaa uudenlaisen tavan välittää tietoa paitsi ihmisten myös tietokoneiden välillä. Näkemystä tukevat myös Jeong et al. (2009), joiden mukaan rakennusten tietomallintaminen on muuttamassa rakennusala suuntaan, jossa suunnitteludokumentteja hyödyntävät myös tietokoneohjelmat.

Kolmantena ja monimutkaisimpana tapana määritellä rakennuksen tietomalli on Succarin (2009) esittämä kolmeen vaiheeseen jakautuva tietomallivisio, jota on havainnollistettu kuvassa 1. Sen ensimmäisessä vaiheessa eri suunnittelualat ja toimijat luovat omat alakohtaiset tietomallinsa, joita käytetään etupäässä automaattisten dokumenttien luontiin sekä kolmiulotteisten havainnekuvien tuottamiseen. Toisessa, edistyneemmässä vaiheessa, suunnittelualat kykenevät yhdistelemään alakohtaisia mallejaan ja pyrkivät suunnittelualojen rajat ylittävään yhteistyöhön. Kolmannessa ja kehittyneimmässä vaiheessa tietomalleista tulee monialaisia malleja, joita ylläpidetään ja hyödynnetään kaikissa rakennuksen elinkaaren vaiheissa. Succarin (2009) vision mukaan viimeisessä vaiheessa tapahtuu kokonaisvaltainen integroituminen, jossa ihmiset, tietojärjestelmät ja liiketoiminta hyödyntävät kaiken käytettävissä olevan tietotaidon rakennusprojektin onnistumiseksi ja maksimoidakseen tehokkuuden kaikissa rakennustuotannon vaiheissa.



Kuva 1. Laatikot havainnollistavat Succarin (2009) tietomallivisio integroitumisen kolmea vaihetta, jotka vasemmalta oikealle ovat nimeltään mallinnus (engl. modelling), yhteistyö (engl. collaboration) ja integraatio (engl. integration).

Succarin (2009) esityksen kaltaista koko rakennustuotannon kattavaa tietomallia Howell & Batcheler (2005) pitävät utopiana, vaikka tunnustavat tietomallintamisen potentiaalin hyödyn erityisesti monialaisessa tiedonsiirrossa. Myös Coenders (2010) epäilee rakennuksen tietomallintamisen kokonaisvaltaista integroitumista ja esittää, että tietomalli visiona ja teknologiana tulee erottaa toisistaan. Hänen mielestään on helppo todeta tietomallin visioon liittyvä potentiaalinen hyöty, kuten vähentyvät suunnittelu- ja rakennusvirheet tai tarkentuvat kustannusarviot, mutta todellisuudessa tietomallitekniikan nykytila on kaukana edellä esitetystä visiosta. On jopa mahdollista, että nykyiset sovellukset perustuvat liian yksinkertaisiin periaatteisiin, jotta vision tekninen saavuttaminen olisi edes mahdollista (Coenders 2010).

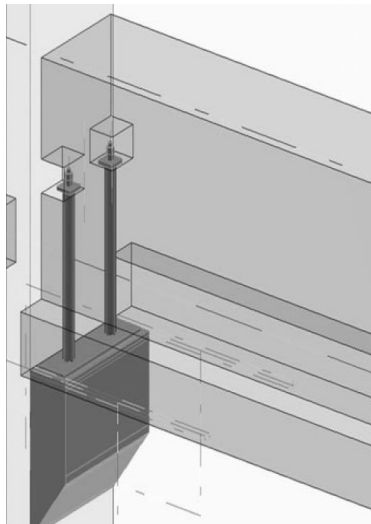
Edellä on esitetty kolme tapaa määritellä rakennuksen tietomalli. Niiden perusteella voidaan todeta, että rakennuksen tietomallia voidaan lähestyä monesta eri näkökulmasta. On vaikeaa antaa yksiselitteistä määritelmää rakennuksen tietomallille. Toisaalta se on ohjelmistotekniikkaan perustuva teknologia, toisaalta taas kokonaan uudenlainen tapa ajatella ja toteuttaa rakennussuunnittelua. Määritelmissä on havaittavissa myös se, että tietomallipohjainen rakennussuunnittelu on murrosvaiheessa, jossa on edelleen käytössä perinteisiä dokumenttipohjaisen suunnittelun keinoja, mutta samalla ollaan ottamassa käyttöön uutta teknologiaa ja uudenlaisia tapoja toteuttaa suunnittelua.

2.2.2 Tietomalli oliotietokantana ja sen näennäinen älykkyys

Tietomallintamisen yhteydessä käytetään toisinaan myös käsitteitä oliopohjainen mallintaminen, oliopohjainen suunnittelu tai älykäs mallintaminen (Kymmell 2008, 31). Oliot ovat olio-ohjelmoinnissa ohjelmistorakenteen perusyksiköitä, joista ohjelmakokonaisuus koostuu. Rakennuksen tietomallissa eri rakennusosat ovat olioita. Oliot voivat sisältää tiedon lisäksi myös toiminnallisuutta, joka voi määrittää rakenneosan eli olion käyttäytymistä. Toiminnallisuutensa avulla oliot kykenevät olemaan vuorovaikutuksessa keskenään. Ne voivat tarkkailla muiden olioiden ominaisuuksia, ja toimia käyttäjän antamien reunaehtojen mukaan, jos tarkkailukohteen ominaisuuden muuttuvat. On kenties liioiteltua sanoa olioita älykkäiksi, mutta perusteluta on sanoa, että niillä on älykkyyden kaltaisia piirteitä.

Oliopohjaisuutta rakennuksen tietomallissa voidaan valaista yksinkertaisella esimerkillä, jossa tarkastellaan palkkia ja pilaria. Pilari, tai tarkemmin sanottuna pilariolio, voidaan määritellä sijoittumaan palkkiolion alapinnan keskikohtaan. Pilarille siis asetetaan riippuvaisuus eli relaatio palkin pituudesta. Mikäli palkin pituus muuttuu, myös pilarin sijainti automaattisesti muuttuu. Leikkimielisesti voidaan ajatella, että rakenneosat reagoimalla ympäristöönsä kykenevät elämään. Tässä mielessä suomenkielinen nimitys olio on osuvampi verrattuna englanninkieliseen vastineeseensa *object*, joka tarkoittaa suomeksi esinettä tai kappaletta.

Mitä enemmän rakenneosiin sisällytetään tietoa ja toiminnallisuutta, sitä enemmän osilla ja tietomallilla voidaan ajatella olevan älykkyyden piirteitä. Tiedon sisällyttämiseen liittyy kuitenkin ongelma, jota on havainnollistettu kuvassa 2. Huomataan, että melko yksinkertaisen elementtipalkin ja -pilarin liitoksen mielekkääseen ominaisuuksien ja relaatioiden kuvaamiseen tarvitaan pitkälle yli sata muuttujaa, eli parametria (Lee et al. 2005). Tästä seuraa helposti ristiriitoja eri muuttujien tai relaatioiden välillä. Ristiriitojen ratkaiseminen jää tyypillisesti ihmisen tehtäväksi, koska ohjelmat eivät toistaiseksi ole riittävän kehittyneitä, jotta ne kykenisivät itse ratkaisemaan ongelman tai edes ehdottamaan ratkaisua ongelmaan.



Vari...	Formula	Variable value	Variable type	Value type	Variable visi...	Variable label in dialog
P1	1"	1"	Parameter	length	True	joint
D1	=P1	1"	Distance	length	False	D1.FITTING.Web right plane
D2	=P2	1/2	Distance	length	False	D2.CORBEL.Lower flange bott...
D3	=P2	1/2	Distance	length	False	D3.CORBEL.Lower flange bott...
P2	1/2	1/2	Parameter	length	False	BEARING JOINT
D4	0"	0"	Distance	length	False	D4.CORBEL.PLATE.End plane
D5	0"	0"	Distance	length	False	D5.CORBEL.PLATE.End plane
D6	1/2	1/2	Distance	length	False	D6.bearing pad.End plane
D7	1/2	1/2	Distance	length	False	D7.bearing pad.End plane
D8	0"	0"	Distance	length	False	D8.linecut.Bottom plane
D9	0"	0"	Distance	length	False	D9.linecut.Top plane
D10	0"	0"	Distance	length	False	D10.linecut.Bottom plane
D11	0"	0"	Distance	length	False	D11.linecut.Top plane
D12	0"	0"	Distance	length	False	D12.linecut.Top plane
D13	0"	0"	Distance	length	False	D13.linecut.Bottom plane

D108	3"1/2	3"1/2	Distance	length	False	D108.BEAM.BLOCKOUT.Top ...
D109	0"	0"	Distance	length	False	D109.NUT.Top plane
D110	3/4	3/4	Distance	length	False	D110.NUT.Top plane
D111	3/4	3/4	Distance	length	False	D111.NUT.Top plane
D112	1"1/2	1"1/2	Distance	length	False	D112.NUT.Top plane
D113	2"	2"	Distance	length	False	D113.ROD.Top plane
D114	1/2	1/2	Distance	length	False	D114.BEAM.SLEEVE.Top plane
D115	1/2	1/2	Distance	length	False	D115.BEAM.SLEEVE.CUT.To...
D116	1/2	1/2	Distance	length	False	D116.BEAM.SLEEVE.Top plane
D117	1/2	1/2	Distance	length	False	D117.BEAM.SLEEVE.CUT.To...
D118	0"	0"	Distance	length	False	D118.CORBEL.Web right plane
D119	0"	0"	Distance	length	False	D119.CORBEL.Web right plane

Kuva 2. Vasemmalla on suhteellisen yksinkertainen ja tyypillinen betonielementtipalkin liitos runkorakenteeseen, ja oikealla on esitetty osa liitokseen liittyvistä parametreista. Ääri vasemmalla olevasta sarakkeesta voidaan erottaa, että parametreja on määritelty 119 kappaletta. Yhteensä liitoksen määrittelemiseen tarvittiin 160 parametria. (Lee et al. 2005).

2.3 Hyödyt ja edut rakennesuunnittelussa

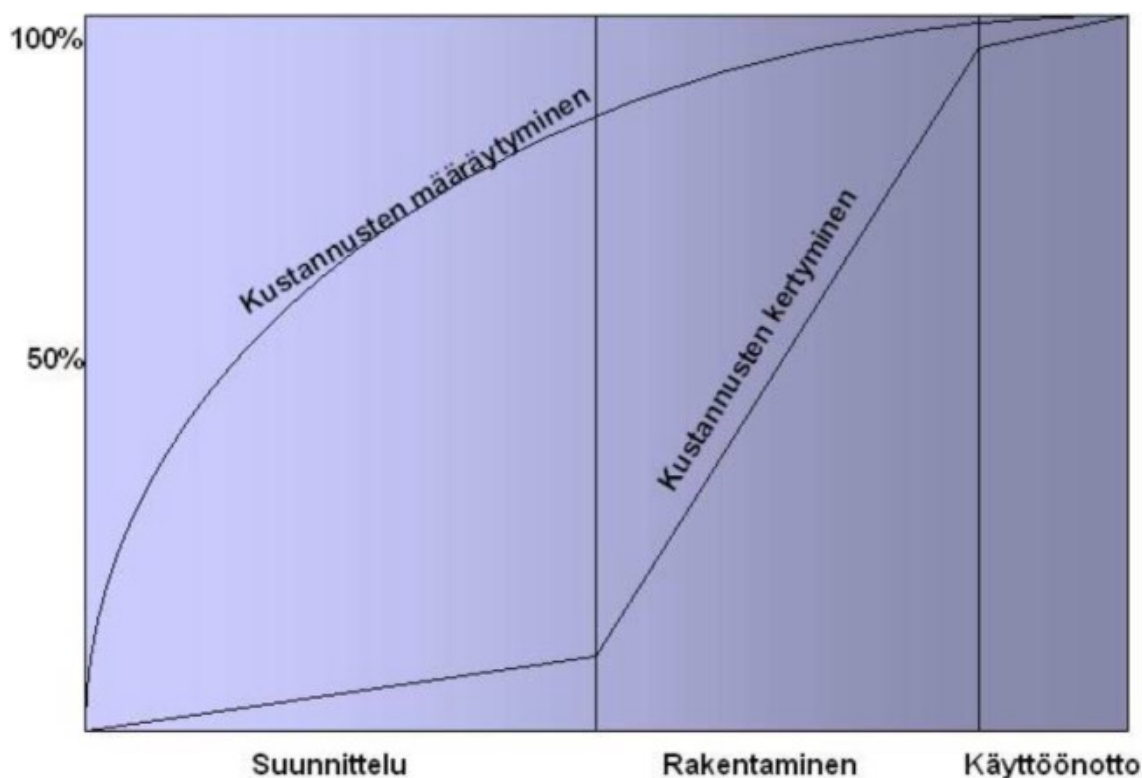
Vertailemalla rakennushankkeen suunnitteluvaiheita perinteisessä dokumenttipohjaisessa ja tietomallintavassa projektissa voidaan tietomallin käytöllä saavuttaa tiettyjä etuja. Taulukkoon 1 on koottu Eastman et al. (2011) listaamia tietomallintamisen yleisiä hyötyjä suunnitteluvaiheessa. He tarkastelevat rakennushankkeen suunnitteluvaihetta kokonaisuutena eivätkä alakohtaisesti, mutta kohdat ovat sovellettavissa myös rakennesuunnitteluun. Listaan voisi lisätä Romo et al. (2005) esittämän näkemyksen, jonka mukaan suunnittelutietojen hallinta on tietomallintavassa suunnittelussa helpompaa. Merkitävimpänä hyötynä tietomallintavassa suunnittelussa Sanguinetti et al. (2012) näkevät erilaiset tietomallia hyödyntävät analysointi- ja arviointiohjelmat perinteiseen (vrt. taulukon 1 kohtaa 7). Rakennesuunnittelussa tämä käytännössä tarkoittaa erilaisia tietomallin dataa hyödyntäviä laskentaohjelmia, joiden avulla rakenteiden kuormien ja eri kuormitustapausten vaikutuksia voidaan nopeasti analysoida.

Taulukko 1. Tietomallipohjaisen suunnittelun hyötyjä voidaan tarkastella eri näkökulmista, joista alla on listattu hyötyjä suunnitteluvaiheessa (Eastman et al. 2011, 20–23).

Tietomallintamisen hyötyjä suunnittelulle:

- 1) Suunnitelmavaihtoehtojen aiempi vertailu
- 2) Varhaisempi ja tarkempi suunnitelmien visualisointi
- 3) Helpompi suunnitelmamuutosten seuranta
- 4) Yhdenmukaisten ja tarkkojen 2D-piirustusten tulostus kaikissa projektin vaiheissa
- 5) Suunnittelualojen varhaisempi yhteistyö
- 6) Varhaisemmat ja tarkemmat kustannusarviot
- 7) Kyky linkittää tietomalli erilaisiin simulaatio- ja analyysisovelluksiin

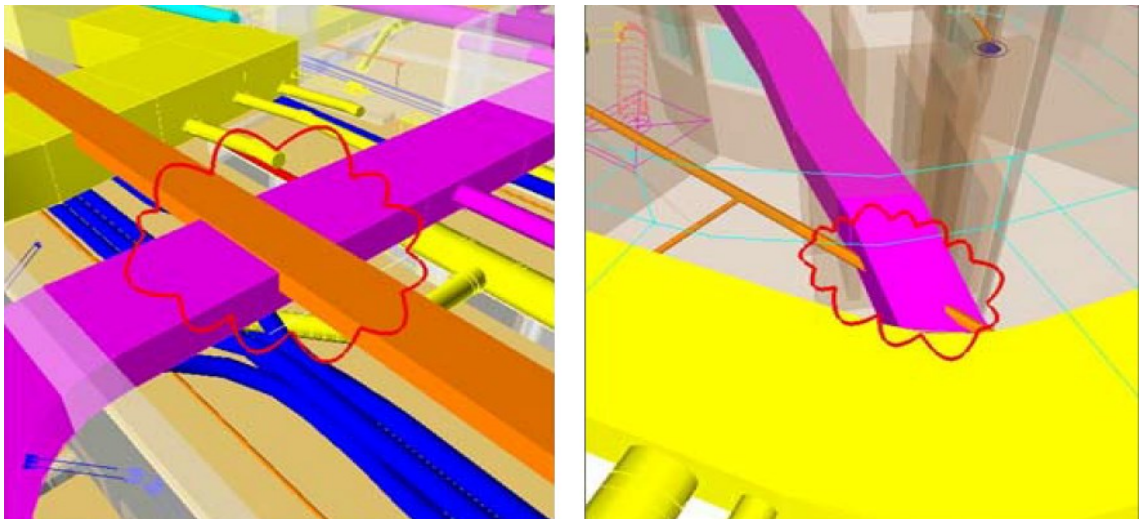
Tietomallintaminen rakennuksen luonnosvaiheessa mahdollistaa varhaisemman arkkitehti- ja rakennesuunnittelun yhteistyön (taulukko 1, kohta 5), mikäli dataa voidaan siirtää eri alojen käyttäminen suunnitteluohjelmien välillä nopeasti ja luotettavasti. Rakennesuunnittelun osalta tämä tarkoittaa, että arkkitehdin luonnostelemaa rakennuksen geometriatietoa voidaan jo suunnittelun alkuvaiheessa hyödyntää laskentaohjelmissa ja sitä kautta päästä aikaisessa vaiheessa käsiksi lopullisiin rakenneratkaisuihin. Kuten kuvasta 3 voidaan havaita, kustannusten määräytyminen on voimakkainta suunnitteluvaiheen alussa. Tästä seuraa se, että esimerkiksi arkkitehti- ja rakennesuunnittelun varhaisella yhteistyöllä voidaan saavuttaa taloudellisia säästöjä. Hellsten (2010) kuitenkin huomauttaa, että vaikka varhaisempi yhteistyö on saatu osittain toteutumaan myös todellisuudessa, on siinä edelleen merkittäviä esteitä, jotka tyypillisesti aiheutuvat sovellusten kyvyttömyydestä hyödyntää toistensa dataa.



Kuva 3. Pystyakselilla on kuvattu rakennuksen kokonaiskustannukset prosentteina (%) ja vaaka-askelilla rakennushankkeen vaiheet. Vaikka kustannukset realisoituvat vasta rakennusvaiheessa, määräytyvät ne silti noin 90 %:sti suunnitteluvaiheessa (Romo & Varis 2004, 5).

Varhainen ja tiivis yhteistyö arkkitehti- ja rakennesuunnittelun välillä voi parantaa myös molempien suunnitelmien laatua erityisesti silloin, kun suunnittelutietoa vaihdetaan tiheään. Tällöin muutoksista aiheutuvat ristiriidat kyetään havaitsemaan varhaisessa vaiheessa. Eastman et al. (2011) näkevät, että eräs merkittävimmistä suunnitelmien laatua parantavista tekijöistä on juuri suunnitteluvirheiden vähentyminen. Perinteisessä dokumenttipohjaisessa suunnittelussa tyypillisiä virheitä ovat virheelliset tai puutteelliset lähtötiedot, sekä väärin oletusten tekeminen epätasmoisten lähtötietojen pohjalta. Virheiden vähentymisen merkittävänä tekijänä on 3D-mallin visuaalisuus ja havainnollisuus, joka Sulankiven (2004) mukaan vähentää erityisesti korkeusasemia koskevia suunnitteluvirheitä. Tämä voidaan todeta myös kuvasta 4, josta on verrattain helppo havaita rakenteiden törmäyminen verrattuna siihen, että asiaa tarkasteltaisiin kaksiulotteisten suunnitelmien avulla. Korpela (2011, 7) näkee kolmiulotteisuuden hyötynä myös helpomman suunnitteluratkaisujen hahmottamisen ja ymmärtämisen.

Merkittävänä tietomallin hyötynä rakennesuunnittelulle pidetään tietotekniikan tuomaa automaatiota (Sulankivi 2004, 16). Tietomallista voidaan tuottaa nopeasti erilaisia listoja ja luetteloita, kuten esimerkiksi paalu-, rauditus- ja elementtiluetteloita. Näkemystä automaation hyödyllisyydestä tukee myös Korpelan (2011) tutkimus, jossa haastatellut Suomessa tietomallin parissa työskentelevät asiantuntijat pitivät eräänä tietomallintamisen hyödyllisimmistä ominaisuuksista tietomallista saatavia automaattisia määrätietoja. Penttilä et al. (2006, 14) tuovat esiin, että erilaisilla automaattisilla tarkastustyökaluilla voidaan helpottaa esimerkiksi arkkitehti- ja rakennesuunnitelmien yhteensovitusta ja tarkastusta, ja sitä kautta paitsi parantavat suunnitelmien laatua, myös nopeuttavat suunnittelu-prosessia.



Kuva 4. Kolmiulotteiset mallit helpottavat suunnitteluvirheiden havaitsemista, kuten törmäystarkasteluja rakenteiden ja talotekniikkaosien välillä.

2.4 Tietomallin mallinnusprosessi

Ennen tietomallin mallinnusprosessin aloittamista, on päätettävä monista mallinnustapoihin liittyvistä asioista. Tapaan ja mallinnustarkkuuteen tarkkuuteen vaikuttavat käytettävät ohjelmat, sillä jotkin ohjelmat saattavat esimerkiksi käsitellä rakennuksen lohkoja tai kerroksia omina tiedostoinaan tai käsitellä rakennusosia kokonaisuuksina (Valjus 2007, 29–30). Tietomallin mallinnustavan valinnoissa pyritään tyypillisesti siihen, että rakennosat mallinnetaan siihen tarkoitetulla työkalulla, jolloin niille muodostuu tyypittävät tunnistet. Esimerkiksi seinät mallinnetaan seinätyökalulla, jotta tietomallia hyödyntävät ohjelmat kykenevät lukemaan tunnisteen, ja käsittelemään rakennosaa seinänä. Kuitenkin myös pienemmillä valinnoilla voi olla merkitystä. Esimerkiksi pilarissa on tyypillisesti kaksi niin sanottua mallinnuspistettä, alku- ja loppupää, joiden suhteen määritellään muu geometria. Näin ollen mallinnettaessa kulmapilari sen uloimmasta kulmapisteestä, ei pilarin koon muuttuminen aiheuta pilarin siirtymistä pois ulkoseinälinjojen risteyksestä.

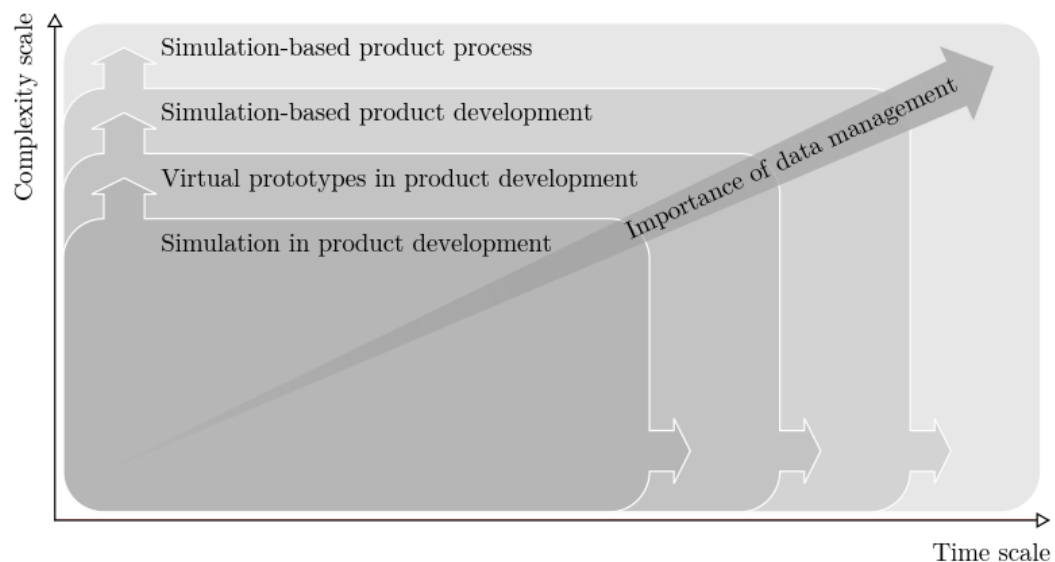
Valjuksen (2007) mukaan tiedon tallentamisen lisäksi tietomallintamiseen liittyy keskeisesti rakennushankkeen eri osapuolten välisen yhteistoiminnan periaatteet:

- miten tietomalli tehdään,
- kuka tietomallin tekee,
- mitä tietoja tietomalliin sisällytetään, sekä
- mitä ja missä tarkoituksessa tietomallin tietoja on tarkoitus käyttää.

On tärkeää, että mallintamistapa ja -tarkkuus sovitaan yhteisesti suunnitteluosapuolien kesken ennen mallinnuksen aloittamista ja että sovitaan yhtenäiset käytänteet siitä, miten rakenneosia mallinnetaan (Valjus 2007, 31).

Tietomallinnuksen luonnosvaiheessa tulee pyrkiä käyttämään yleisiä ratkaisuja ja mallintamaan esimerkiksi liitokset mahdollisimman myöhäisessä vaiheessa, jotta vältetään ”liian valmiin” mallin työläältä muuttamiselta (Valjus 2007, 31). Kolmiulotteisen mallin sujuvan käytön kannalta yksi merkittävistä rajavedoista on, mitkä asiat mallinnetaan ja mitkä asiat sisällytetään attribuuttitietoina eli rakenneosiin liittyvänä tietona. Mitä enemmän rakenneosia ja niiden yksityiskohtia mallinnetaan, sitä enemmän mallin pyörittäminen vaatii tietokoneelta suoritustehoa.

Tällä hetkellä rakennushankkeen suunnitteluvaiheessa tietomallia hyödynnetään tyypillisesti siten, että eri suunnitteluosapuolien työ tapahtuu itsenäisesti ja suunnitelmia sovitaan yhteen sopivin määräajoin sitä mukaan, kun suunnittelu etenee. Suunnittelun edetessä tietomallin sisältämä tietomäärä kasvaa, kun malliin lisätään tietoa eri tiloista ja rakenteista sekä niiden ominaisuuksista, mitoista ja määristä. Tämä asian vaikutuksia on Kortelainen (2011) esittänyt kuvaajassaan kuvassa 5. Sitä soveltaen voidaan todeta, että ajan kuluessa mallin sisältämä tietomäärä kasvaa ja samalla kasvaa myös mallin monimutkaisuus. Tämä puolestaan lisää tiedonhallinnan merkitystä tietomallinnusprojektissa. Penttilä et al. (2006, 36) esittävät, että tiedon ja tiedonsiirron koordinoiminen on tärkeää, koska siinä esiintyy tyypillisesti paljon ongelmia, ja yhteisesti sovitulla tiedonhallinnalla on mahdollista nopeuttaa suunnitteluprosessia



Kuva 5. Kortelaisen (2011) esittämässä kuvaajassa tuoteprosessin kehittyessä ajan myötä kasvaa myös simuloitavan mallin monimutkaisuus sekä tiedonhallinnan merkitys.

3 Laskentamallin hyödyntäminen tietomallintavassa rakennesuunnittelussa

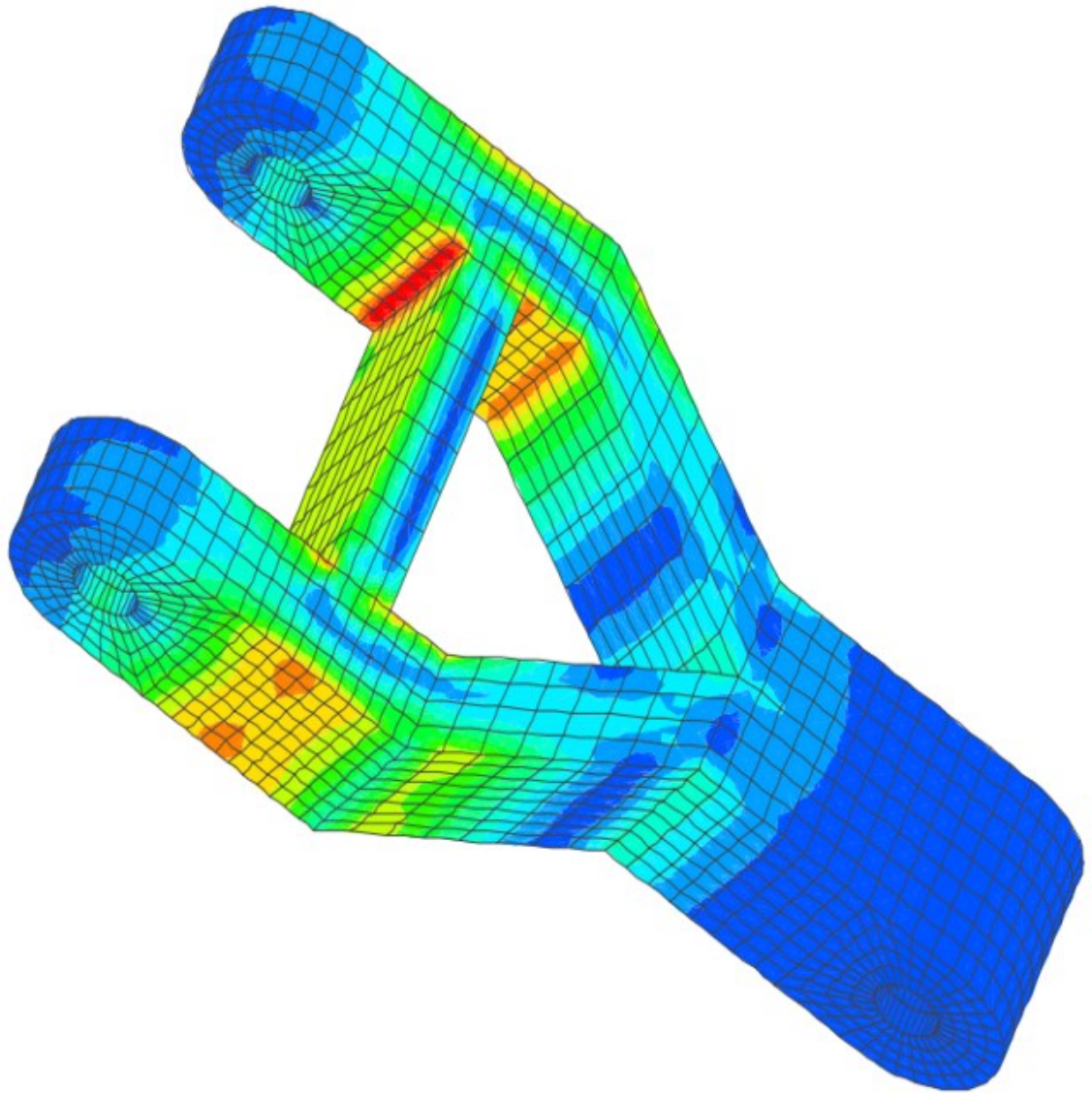
3.1 Laskentamallin matemaattinen ymmärtäminen ja perusteet

Kuten edellä on esitetty, merkittävänä osana tietomallipohjaista rakennussuunnittelua ovat erilaiset tietomallin dataa hyödyntävät laskenta- ja analyysiohjelmat. Ohjelmien kirjo on laaja ja ne ovat tyypillisesti pitkälle omaan aihepiiriinsä erikoistuneita. Niillä voidaan mallintaa muun muassa rakenteiden rakennusfysikaalista toimintaa tai tuulen käyttäytymistä rakennusten julkisivujen pinnoilla. Yhteistä näille ohjelmille on, että niissä pyritään esittämään jokin todellisuuden fysikaalinen ilmiö matemaattisin mallein, joita tutkimalla voidaan saada lisätietoa ilmiön vaikutuksista. Kouhia & Tuomala (2009) esittävät, että fysikaalisten ilmiöiden mallintamisessa päädytään tyypillisesti osittaisdifferentiaaliyhtälöiden muodostamaan yhtälösystemiin, joka rakentuu kolmentyyppisistä relaatioista:

- (1) luonnonlait,
- (2) materiaalinen käyttäytyminen (konstitutiiviset yhtälöt), ja
- (3) kinematiikka (geometrinen liikeoppi).

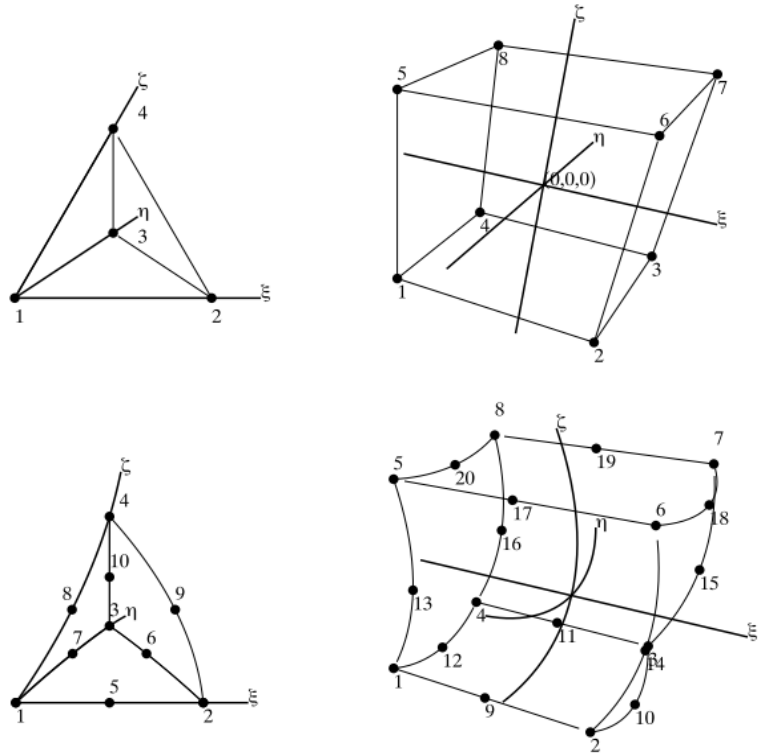
Todellisuuden ilmiöistä esitetyt matemaattiset mallit ovat usein niin monimutkaisia, että analyttinen ratkaisu on käytännössä mahdotonta. Tällöin käytetään numeerisia menetelmiä, joiden tavoitteena on löytää ongelmaan likimääräisratkaisu, jonka voidaan katsoa tyydyttävällä tarkkuudella vastaavan todellisuutta. Numeerisia menetelmiä ovat esimerkiksi differenssi- ja kontrollitilavuusmenetelmä, mutta rakenneanalyysissä eräs käytetyimmistä on elementtimenetelmä (Hämäläinen & Järvinen 2006, 45). Siitä käytetään myös nimitystä FEM-laskenta, jonka etuliite on akronyymi ratkaisukeinon englanninkielisestä nimestä *finite element method* eli suomeksi äärellisten elementtien menetelmä. Sen suurimpia etuja ovat soveltuvuus monimutkaisten laskenta-alueiden kuvaamiseen (Hämäläinen & Järvinen 2006, 45).

Kuvasta 6 voidaan hahmottaa käytäntöön sovellettuna elementtimenetelmän perusidea: geometrialtaan monimutkainen kappale on jaettu lukuisiin muodoltaan yksinkertaisiin osiin eli elementteihin. Elementtien muodostama kokonaisuutta kutsutaan elementtiverkoksi, ja sen muodostamista verkon generoinniksi tai puhekielessä myös verkottamiseksi. Jokaista elementtiä hallitsevat osittaisdifferentiaaliyhtälöt, jotka muunnetaan algebralliseksi yhtälöryhmiksi, joita soveltamalla saadaan koko systeemiä kuvaava approksimoiva algebrallinen yhtälöryhmä (Hämäläinen & Järvinen, 49–50). Toisin sanoen, tutkimalla yksittäisiä elementtejä ja niiden vuorovaikutuksia toisiin elementteihin, voidaan muodostaa kokonaiskuva fysikaalisesta systeemistä (kappale) luodun matemaattisen mallin käyttäytymisestä, kuten kuvan 6 tapauksessa jännitystilojen jakaumasta kappaleen pinnalla.



Kuva 6. Laskentaohjelma on jakanut kappaleen osiin ja elementtimenetelmän avulla laskenut jännitystilat kappaleen pinnalla. Tuloksia visualisoidaan tyypillisesti värein, kuten tässä tapauksessa punertaville alueille kohdistuu suurin jännitystila ja sinisille pienin. (Lähteenmäki 2009.)

Kolmiulotteisten kappaleiden ratkaisuun käytetään tyypillisesti kuvassa 7 esitettyjä tilavuuselementtejä, kun kaksiulotteisten kappaleiden ratkaisuun käytettisiin vastaavia tasoelementtejä. Kuvassa erottuvat myös numeroidut elementtien solmupisteet, joita voidaan ajatella elementtejä toisissaan kiinni pitävinä liimapisaroina. Elementtien koot sekä muodot riippuvat kappaleen geometriasta ja saattavat vaihdella yksittäisen kappaleen sisällä. Niiden tulee kuitenkin aina täyttää laskenta-alue kokonaan ilman tyhjiä kohtia, eivätkä ne saa olla miltään osin päällekkäin. Elementtien käyttäytyminen tulee olla tasapainossa ja yhteensopivaa viereisten elementtien kanssa (vrt. (2) kinematiikka), jonka lisäksi niiden käyttäytyminen tulee olla materiaalilakien mukaista (vrt. (3) konstitutiiviset yhtälöt).



Kuva 7. Tyypillisiä tilavuuselementtejä ovat neli- ja kuusitahokas (ylinnä), joiden pinnat voivat olla myös kaarevia (alhaalla). Kuvassa erottuvaat myös numeroidut solmupisteet sekä niin sanotut lokaalikoordinaattiakselit. (Haataja et al. 2002, 266.)

Tyypillisesti rakennesuunnittelussa käytössä olevat laskentaohjelmat olettavat, että materiaalit käyttäytyvät elastisesti, eli kuormitusten aiheuttamien siirtymien sekä jännitysten yhteys on lineaarinen. Sovellukset saattavat tarjota plastisen käyttäytymisen tutkimiseen erillisiä työkaluja, tai sitä voidaan mallintaa likimääräisesti iteratiivisella prosessilla, jossa paikallisia materiaaliparametreja muokataan sopivasti vastaamaan kulloistakin kuormitustilaa. Polak & Pande (1999) kuitenkin huomauttavat, että epälineaaristen analyysien tuloksissa on havaittu epäyhdenmukaisuuksia esimerkiksi nurjahdustarkasteiluissa.

Haataja et al. (2002) mukaan elementtimenetelmä on periaatteeltaan helppo ymmärtää ja suoraviivainen toteuttaa, sekä tyypillisissä tekniikan sovelluksissa niin monimutkainen, että laskennan apuna on käytettävä tietokonetta. Rakenneanalyysisovelluksilla mallinnetuihin rakenteisiin kohdistetaan halutut kuormitukset ja asetetaan systeemin reunaehdot. Sovellukset verkottavat rakenteet automaattisesti, mutta suunnittelija voi halutessaan tihentää (tai harventaa) verkkoa. Verkon tihentäminen voi esimerkiksi tulla kysymykseen raskaasti kuormitetulla kriittisellä alueella, koska tiheämmällä verkolla myös analyysin tulokset ovat tarkempia.

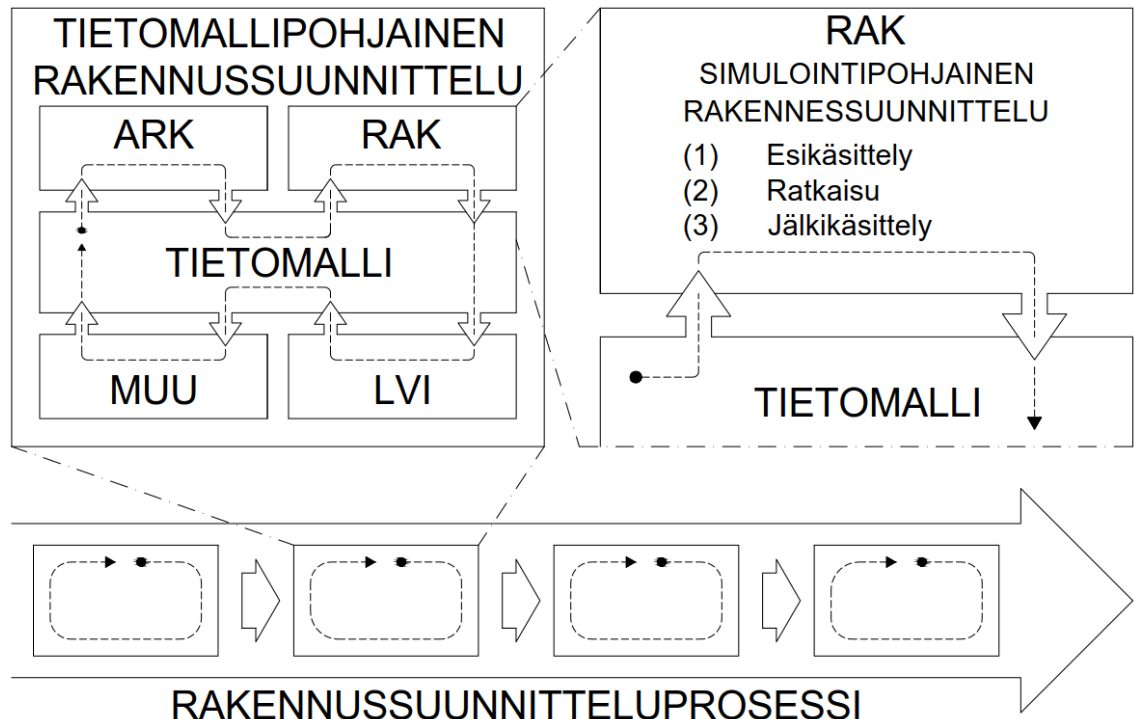
3.2 Simulointipohjainen rakennesuunnittelu

Laskentaohjelmat ovat nykypäivänä ominaisuuksiltaan ja mallinnuskyvyiltään niin edistyneitä, että voidaan puhua simulointipohjaisesta rakennesuunnittelusta. Esimerkiksi Shephard et al. (2004) määrittelevät simulointipohjaisen suunnittelun prosessiksi, jossa simulointi on ensisijainen keino suunnitelmien arviointiin ja todentamiseen. Tätä voidaan verrata siihen, että laskentamallin avulla tutkitaan virtuaalisissa ympäristöissä rakenteiden todellista käyttäytymistä matemaattisten mallien avulla. Ojala (2013) listaa kahdeksan simulointipohjaisen suunnittelun hyödyllistä ominaisuutta, jotka kohtia (1) ja (6) lukuun ottamatta ovat sovellettavissa myös rakennesuunnitteluun:

- (1) Vähentää sekä prototyyppien tarvetta, että fyysisiä testejä.
- (2) Nopeampi suunnitteluprosessi.
- (3) Paremmat suunnitteluratkaisut, koska vaihtoehtoja on tutkittu laajemmin.
- (4) Suunnitteluvirheiden aikaisempi tunnistaminen tuotteen elinkaareissa.
- (5) Tuloksia saadaan nopeammin tukemaan suunnittelupäätöksiä.
- (6) Virtuaaliset prototyypit.
- (7) Helpompi kommunikaation yhtiön sisällä alentaa valmistuskustannuksia.
- (8) Tuotetietoa saadaan enemmän.

Tällaisessa simulointipohjaisessa rakennesuunnittelussa voidaan nähdä toistuvan kolme vaihetta: (1) esikäsittely, (2) ratkaisu ja (3) jälkikäsittely (vrt. Kortelainen 2011, 26). Työläin ja eniten aikaa vievä on esikäsittelyvaihe, jonka tärkeimpänä päämääränä on luoda realistinen ja toimiva laskentamalli. Laskentamallin muodostamisessa pyritään hyödyntämään tietomallista saatavaa dataa. Ratkaisu- ja jälkikäsittelyvaiheessa suoritetaan analysointi sekä arvioidaan laskentamallista saatujen tulosten oikeellisuutta ja luotettavuutta. Lopulta tulosten perusteella tehdään tarvittaessa muutoksia rakenteisiin ja ne muutokset viedään tietomalliin.

Simulointipohjaista rakennesuunnittelua osana tietomallintavaa rakennesuunnittelua on havainnollistettu kuvassa 8. Tietomallintavassa rakennussuunnittelussa eri suunnittelualat voidaan nähdä osaprosesseina, jotka tuottavat tietomalliin tietoa ja sen kautta jakavat sitä eteenpäin muille suunnitteluosapuolille. Simulointipohjainen rakennesuunnittelu yhtenä osaprosessina on itseään tarkentava, iteratiivinen prosessi, jossa suunnitelmat muuttuvat ja tarkentuvat silmukkamaisesti kohti lopullista muotoa suunnittelutiedon kiertäessä suunnittelualojen sisällä ja suunnittelualojen välillä (Niggel et al. 2006).



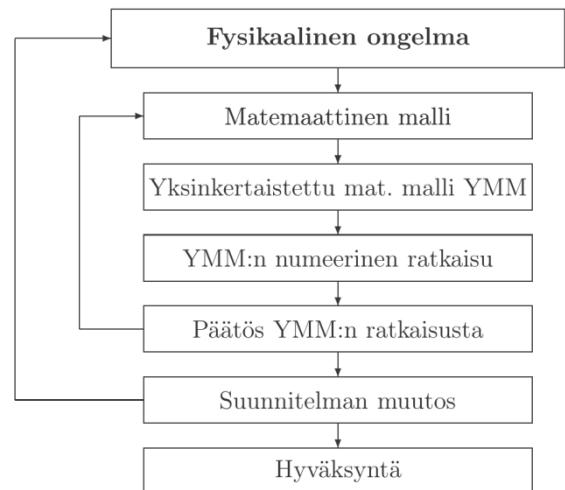
Kuva 8. Rakennesuunnitteluprosessia on ylinnä kuvattu itseään toistavina vaiheina eli silmukoina, joiden aikana suunnittelualojen yhteistyönä tuotetaan suunnittelutietoa rakennuksen toteuttamiseksi. Rakennesuunnittelun ytimessä on tietomalli, joka toimii suunnittelutiedon välittämisen rajapintana suunnittelualojen välillä. Rakennesuunnittelu hyödyntää tietomallin tietoja suunnittelussa, ja rakenneanalyyysien avulla tuottaa uutta tietoa, joka tallennetaan lopulta takaisin tietomalliin.

Kurowski (2004) näkee erilaiset laskentaohjelmat käyttökelpoisina työkaluina, jotka voivat parhaimmillaan avustaa suunnittelupäätösten tekemisessä, nopeuttaa niitä ja viedä suunnittelua eteenpäin. Samalla hän kuitenkin huomauttaa, että erityisesti suunnitteluvaiheen alussa tarvitaan osaamista, jotta tulosten pohjalta voidaan tehdä luotettavia tulkintoja (Kurowski 2004). Tämän seurauksena rakennesuunnittelun toimenkuva on muuttunut analyttisempaan suuntaan, jossa on korostunut rakenteiden toiminnallista ymmärtämistä. Coenders (2007) käyttää asiasta nimitystä käsitteellinen rakennesuunnittelu (engl. *conceptual design of structures*). Toisin sanoen laskentamallin käyttö vaatii, että suunnittelija kykenee mallintamaan rakenteet ja asettamaan niiden reunaehdot siten että ne vastaavat todellisuutta. Tämän lisäksi suunnittelijan on ymmärrettävä ohjelman laskentaperiaatteet ja osattava tehdä tulosten perusteella luotettavia tulkintoja kuormien vaikutuksista.

Lopullisten rakenneratkaisujen löytäminen varhaisessa vaiheessa hanketta voi merkittävästi alentaa rakennuksen kokonaiskustannuksia, minkä vuoksi rakennesuunnittelu pyrittään ottamaan mukaan suunnitteluprosessiin mahdollisimman aikaisin. Kuitenkin suunnitteluvaiheen alussa hankekohtainen informaatio on pienimmillään, mikä tekee rakenneratkaisujen valinnasta haasteellista. Laskentaohjelmalla eri vaihtoehtoja voidaan aiempaa nopeammin mallintaa ja analysoida, mikä helpottaa vertailua niiden välillä. Rakenteet on mallinnettava riittävän karkeasti, jotta suunnitteluprosessi pysyy sujuvana, mutta kuitenkin tarkkuudella, jolla analyysistä saadaan luotettavia ja suuntaa-antavia tuloksia. Tämä edellyttää rakennesuunnittelijalta asiantuntevaa näkemystä rakenteiden toiminnasta sekä valistuneita arvauksia.

3.3 Laskentamallin luonti ja analysointi

Aivan kuten Kouhian & Tuomalan (2009) esittämässä fysikaalisen ongelman matemaattisessa ratkaisuprosessissa, joka on esitetty kuvassa 9, myös laskentamallia hyödyntävässä rakennesuunnittelussa on tunnistettavissa vastaavat vaiheet. Rakennesuunnittelussa fysikaalisena ongelmana on tutkittava rakenne ja kuormitusten jakaantuminen sen osissa. Rakenteesta luodaan ohjelmien avulla matemaattinen malli laskentamalli. Tyypillisesti sitä yksinkertaistetaan jättämällä pois rakenteiden yksityiskohtia, joiden vaikutukset ratkaisuun ovat merkityksettömiä. Ohjelman laskettua kuormien jakaantuminen arvioidaan, olivatko luodun matemaattisen mallin lähtötiedot ja oletukset oikein ja ovatko tulokset realistisia. Tarvittaessa tehdään muutoksia matemaattiseen malliin ja laskentaprosessi aloitetaan uudelleen. Mikäli taas tulkitaan tulosten perusteella, että matemaattinen mallin on ollut riittävällä tarkkuudella oikein todellisuuteen verrattuna, mutta suunniteltu rakenne on ollut riittämätön, tehdään muutoksia itse fysikaaliseen ongelmaan eli rakenteeseen.

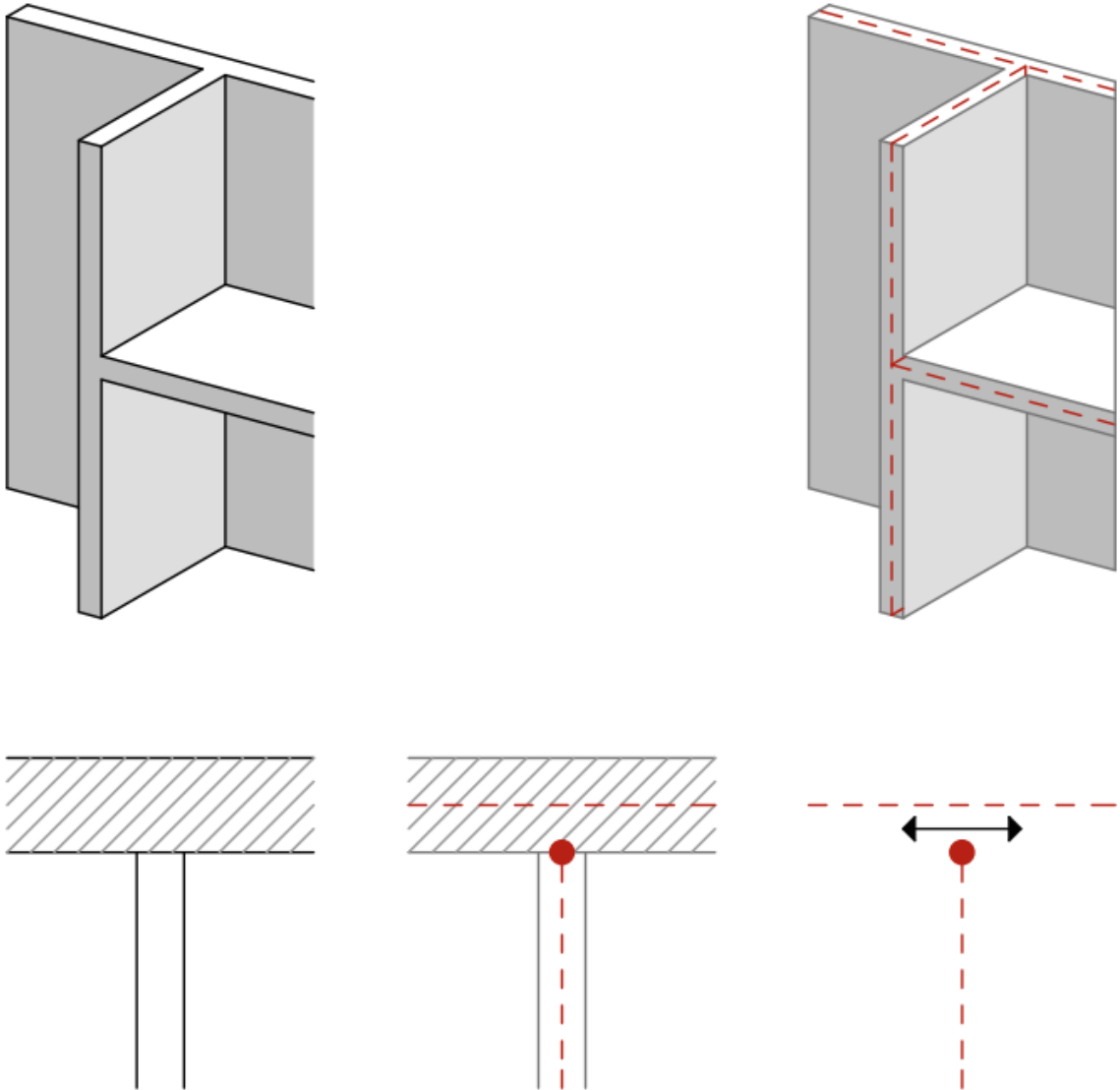


Kuva 9. Vuokaaviossa on tyypillinen tapa saavuttaa matemaattisen mallin ratkaisu. Mallin toimivuutta arvioidaan ja siitä tehdään päätelmiä kyseiseen ongelmaan nähden, minkä jälkeen siihen voidaan tehdä muutoksia. Ratkaisu voi tuoda esiin myös muutostarpeita itse fysikaalisen ongelman määrittelyyn. (Kouhia & Tuomala 2009, 2.)

Laskentamallin luominen voidaan aloittaa suoraan laskentaohjelmassa, mutta tyypillisesti luonnissa hyödynnetään tietomallissa olevaa dataa. Tähän on olemassa kaksi syytä. Ensinnäkin tietomalli sisältää uusimman ja ajantasaisimman käytettävissä olevan tiedon. Toinen syy on, että laskentaohjelmien mallintamisominaisuudet ja -työkalut ovat yksinkertaisempia tietomallintamisovelluksiin verrattuna, joten monimutkaisia malleja on työlästä mallintaa niissä (Coenders 2007). Ideaalitilanteessa laskentamallin tarvitsemat tiedot saadaan suoraan siirrettyä tietomallista ilman välivaiheita, mutta käytännössä tämä erittäin harvoin toteutuu. Eastman (2011) toteaa, että lähes poikkeuksetta laskentaohjelmat vaativat tietomallissa tehtäviä laajoja geometriatietojen valmisteluja, materiaalitietojen määrittelyjä ja kuormitustietojen sovittamista ennen kuin laskentasovellukset voivat luotettavasti käsitellä tietomallin dataa.

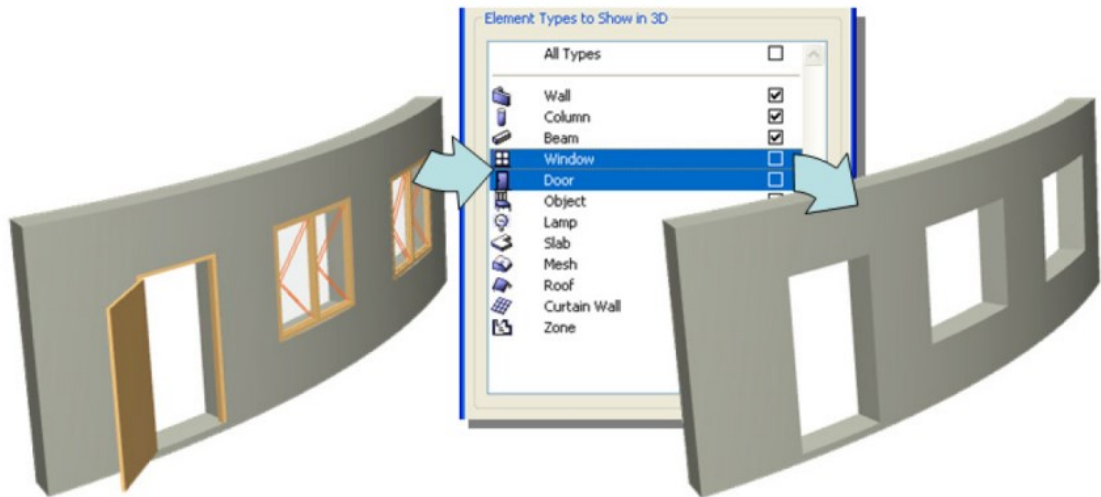
Kurowski (2004) mukaan riisuminen ja täydentäminen ovat kaksi tyypillisintä toimenpiteitä, jotka laskentamallille on tehtävä ennen kuin sitä voidaan alkaa analysoida. Riisumisella tarkoitetaan, että tietomallista siirrettävästä datasta rajataan pois mitoittamisen kannalta epäoleellista tietoa. Täydentämisessä puolestaan yksittäisiä rakenteita muutetaan tai jatketaan, jotta rakenne toimii laskentaohjelman käytänteiden mukaan oikein. Nämä toimenpiteet voidaan tehdä tieto- tai laskentaohjelmassa, mutta tyypillisesti riisuminen tehdään jo tietomallista ennen varsinaisen datan siirtämistä.

Täydentämisen syytä on havainnollistettu kuvassa 10, jossa on esitetty seinien liitoskohta. Laskentamallissa seinien liitoskohdassa keskilinjoihin pitkin kulkevien analyysiviivojen tulee kohdata, jotta ohjelma ymmärtää rakenteen olevan yhtenäinen muodostaen rakenteiden välille solmupisteen. Ohjelmat saattavat kyetä tunnistamaan vastaavia solmujen risteyskohtia automaattisesti, tai niitä varten voi olla erillinen työkalu. Käytännössä työkalujen tunnistamismekanismi ei ole aukoton, ja työkalujen käyttö vaatii tarkkuutta, jotta sovellus täydentää laskentamallia tavalla, joka ei aiheuta vääristymiä laskentaan. Tämä ero havainnollistaa myös tieto- ja laskentamallien geometrian esittämisen erilaisia vaatimuksia (Kurowski 2004). Tietomallissa on luontevaa, että seinä mallinnetaan päättyväksi toisen seinän pintaan eikä sen keskilinjaan.



Kuva 10. Ylhäällä on kolmiulotteinen esitys kahden seinän ja laatan muodostamasta rakenteesta ja alarivillä on esitetty tasokuva seinien liitoksesta. Tyypillinen analyysimallin ja geometriamallin ero havainnoilstuu alarivillä, jossa punaisella katkoviivalla kuvatut analyysiviivat eivät kohtaa. (Steel et al. 2010.)

Kuvassa 11 on visuaalisesti esitetty riisumisen periaate. Tietomallin sisältämästä seinärakenteesta suodatetaan (engl. *filtering*) pois ylimääräiset rakenteet, kuten ikkunat ja ovet, ennen sen siirtämistä laskentamalliin. Osien suodattaminen perustuu rakenneosien luokkiin ja attribuutteihin, sekä tarvittaessa muihin rakenneosien yksilöllisiin tunnuksiin. Luotettava ja sujuva suodattaminen edellyttää, että mallinnuksessa noudatetaan tiukkaa kuria rakenneosien sijoittamisessa asianomaisten tunnusten alle. Lisäksi se vaatii, että mallinnusta tehtäessä on huomioitu niin rakennesuunnittelun kuin muidenkin suunnittelualueiden tarpeet tunnusten luonnissa. Ideaalitilanteessa suodattaminen tehdään jo tietomallissa, mutta käytännössä rakenteita joudutaan toisinaan riisumaan (poistamaan) myös laskentamallissa.



Kuva 11. Ikkunat ja ovet ovat rakenneanalyysin kannalta epäoleellista tietoa, joten ne suodatetaan tietomallisovelluksessa pois ennen mallin siirtämistä laskentaohjelmaan. (Graphisoft 2012, 20.)

Kaikkien näiden toimenpiteiden suorittaminen kuuluu rakennesuunnittelun toimenkuvaan, ja vaatii paitsi kykyä ymmärtää rakenteiden mekaniikkaa myös tuntea rakenteiden yksityiskohtiin liittyvät kansalliset säädökset (Rammant & Adriaenssens 2008). Senescu et al. (2006) esittävät eräänä vaihtoehtona sen, että rakennesuunnittelijan on osallisena mallinnusprosessissa, koska rakennesuunnittelun kannalta on oleellista paitsi mitä on mallinnettu, myös miten malli on mallinnettu. On usein työläämpää muuttaa mallia jälkikäteen, kuin tehdä se ensimmäisellä kerralla oikein. Toisin sanoen mallinnuksessa voidaan vaatia myös tietämystä siitä, miten rakenteita matemaattisesti kohdellaan ja miten ne ovat hyvä laskentamallin kannalta mallintaa.

Eastman (2011) edellyttää tietomallin ja laskentaohjelman tehokkaalta käytöltä kolmea asiaa.

- (1) Tietomallisovelluksesta saatavien tietojen ja relaatioiden tulee olla yhteensopivia ja käyttökelpoisia laskentasovelluksessa haluttua analyysia varten.
- (2) Tietomallin sisältämästä datasta tulee voida koota laskentamalli, joka riittävällä tarkkuudella vastaa todellisuutta, jotta analyysin tuloksia voidaan käyttää suunnittelussa hyväksi.
- (3) Tieto- ja laskentamalliohjelmilla tulee olla yhteisesti tuettu tiedostomuoto, jotta dataa voidaan luotettavasti siirtää ohjelmien välillä. Lisäksi tiedonsiirron tulee sisältää tunnistetietoa, jolla voidaan ylläpitää tietomallista tai laskentamallista tulevia päivityksiä.

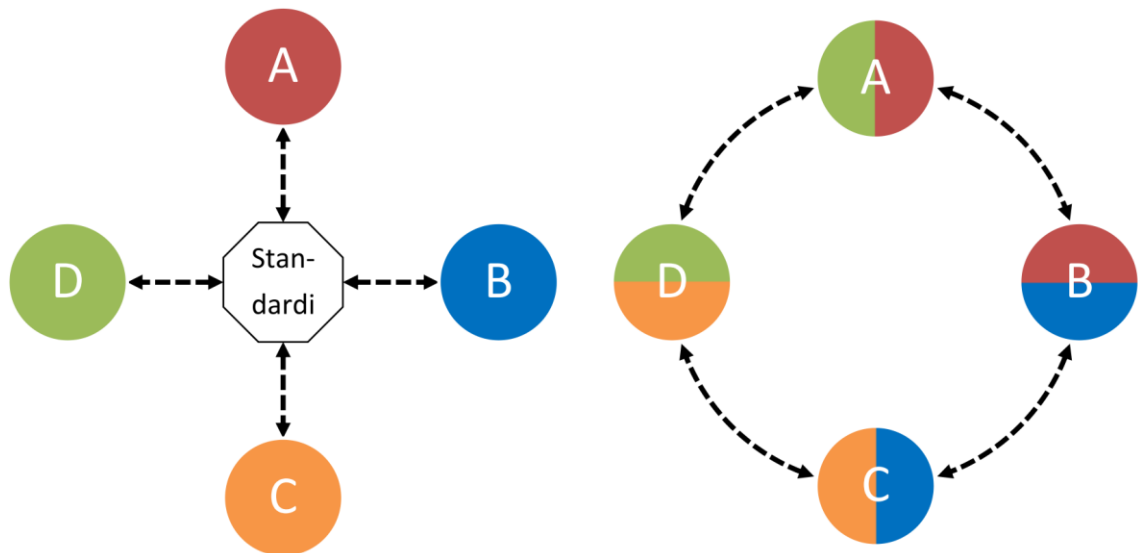
Yleinen vaatimus rakenneanalyysin suorittamiselle on, että siirrettyjen rakenteiden on oltava verkottuvia siten, että analyysisovellus kykenee ratkaisemaan halutun asian, kuten esimerkiksi rakenteiden siirtymät, tyydyttävällä tarkkuudella ja kohtuullisessa ajassa (Kurowski 2004). Tietomallin datan hyödyntämiselle laskentamallissa oman haasteen asettaa tiedonsiirto, joka tulee onnistua siirtämään halutut rakenteet mallien välillä oikein, käyttäjän haluamalla tavalla ja luotettavasti.

3.4 Yhteentoimivuus ja tiedonsiirtoprosessit

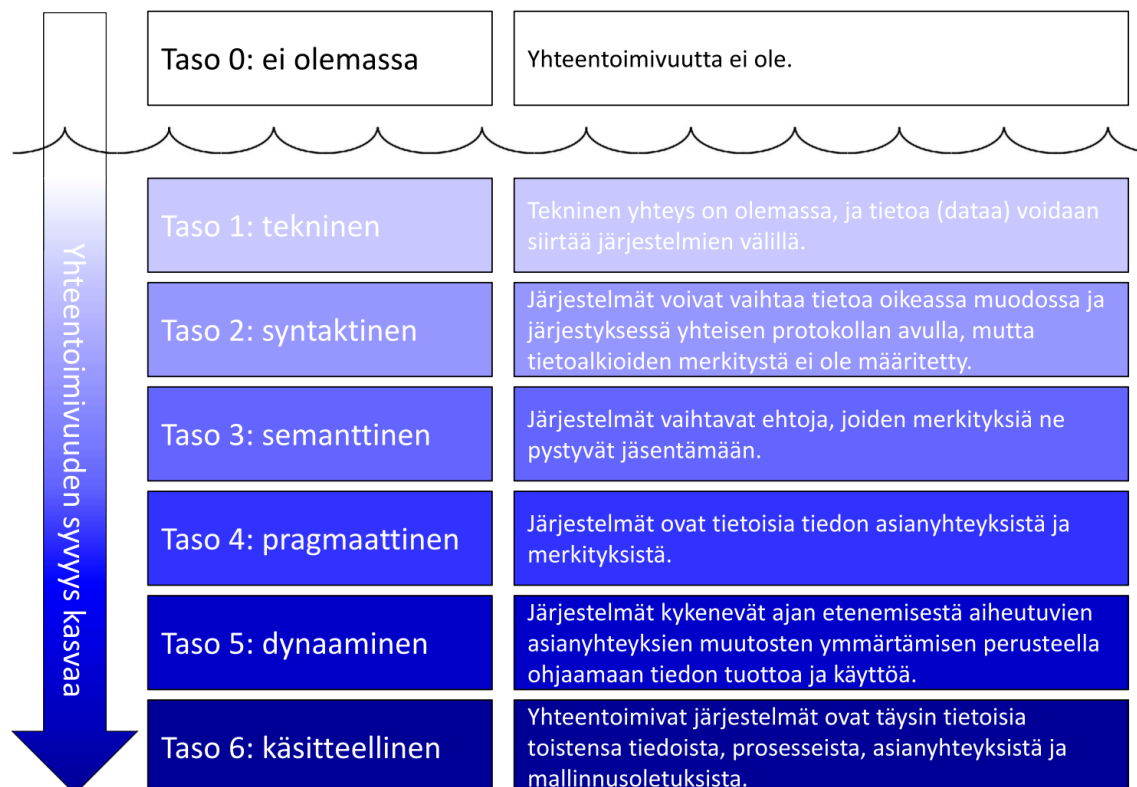
3.4.1 Yhteentoimivuuden käsite ja sen tasot

Yhteentoimivuus (engl. *interoperability*) voidaan tietotekniikassa määritellä tarkoittamaan tietojärjestelmien kykyä viestiä keskenään sellaisella tavalla tai siinä laajuudessa, että ne voivat rutiininomaisesti käyttää toistensa tietoja (TSK 2012). Toinen aihealueeseen liittyvä termi on yhteensopivuus (engl. *compatibility*), jolla tarkoitetaan samassa ympäristössä olevien järjestelmien, laitteiden tai komponenttien kykyä toimia yhdessä ilman, että niitä täytyy muuttaa merkittävästi (TSK 2012). Kuva 12 havainnollistaa termejä, ja siitä voidaan nähdä termien merkittävin ero: yhteentoimivuus ei edellytä samaa toimintaympäristöä. Toisinaan kirjallisuudessa nämä kaksi käsitettä sekoitetaan toisiinsa (esim. Koivu 2002; Rammant & Adriaenssens 2008).

Yhteentoimivuudessa voidaan siirtyvän tietosisällön perusteella erottaa monia tasoja. Kuvassa 13 on esitetty Wang et al. (2009) jaottelu sanallisine kuvauksineen. Heidän jaottelussaan (tekninen) yhteentoimivuus saavutetaan tasolla yksi, jossa datan siirtäminen järjestelmien välillä tulee mahdolliseksi. Tiedonsiirron mahdollisuudet ja tietosisällön syvällisyys lisääntyvät tasoittain, kunnes syvimmällä, käsitteellisellä tasolla, järjestelmät ovat täysin tietoisia toisten järjestelmien tiedoista, prosesseista, asiansyhteyksistä ja mallinnusoletuksista (Wang et al. 2009). Tieto- ja laskentamallien yhteentoimivuudessa rakennussuunnittelussa liikutaan arviolta tasoilla yksi ja kaksi, sillä kuten Hellstenkin (2010) artikkelissaan toteaa, tietomallin tietoja ei kyetä aukottomasti siirtämään eri sovellusten välillä.



Kuva 12. Vasemmalla on havainnollistettu yhteentoimivuutta ja oikealla yhteensopi-
vuutta; ympyrät kuvaavat järjestelmiä ja värit toimintaympäristöjä. Vasemmalla järjes-
telmät toimivat vain omissa toimintaympäristöissä, mutta kykenevät yhteisen standardin
avulla toimimaan yhdessä. Oikealla kukin järjestelmä toimii kahdessa toimintaympäris-
tössä, mutta yhteentoimivuuden puuttuessa kykenevät toimimaan yhdessä vain vierekkäis-
ten järjestelmien kanssa.



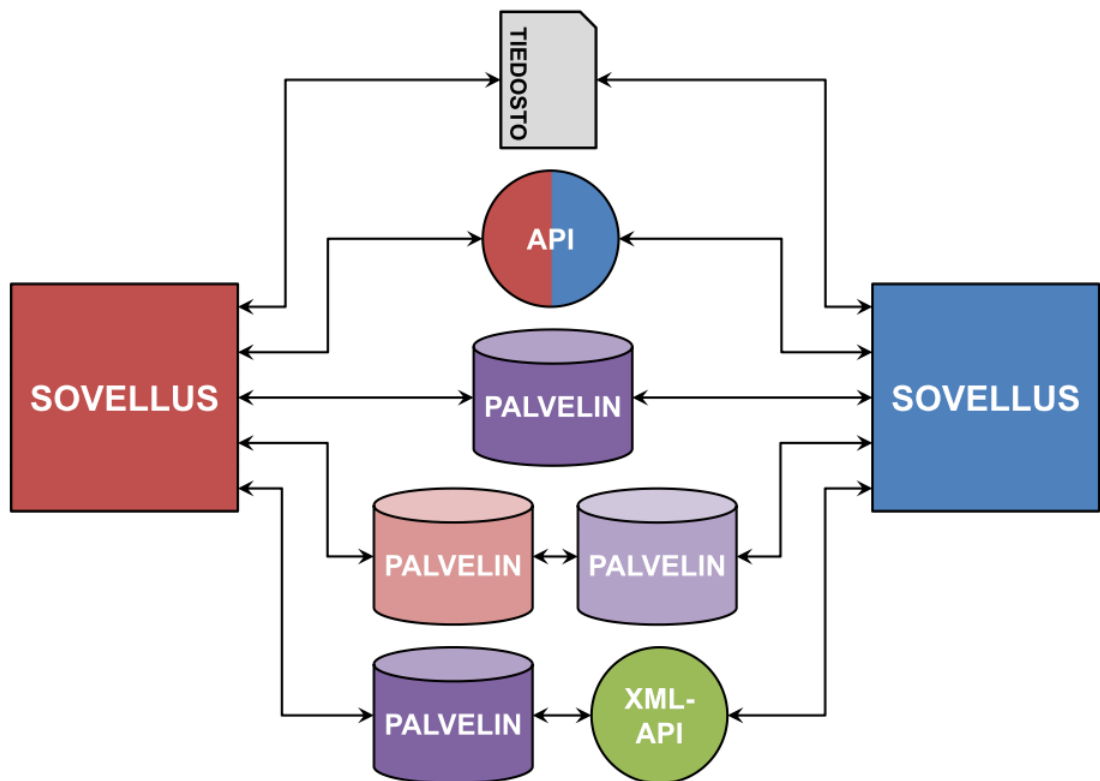
Kuva 13. Käsitteellisen mallinnuksen ja yhteentoimivuuden tasot (engl. levels of concep-
tual modelling and interoperability) esitettyinä Wang et al. (2009) mukaan. Yhteentoimi-
vuuden "syvyyden" kasvaessa lisääntyy myös järjestelmien kyky ymmärtää toisiaan.

3.4.2 Tiedonsiirto yhteisen standardin avulla

Kun teknisen tason yhteentoimivuus on saavutettu, voidaan tiedonsiirto toteuttaa monin eri tavoin. Kuvassa 14 on esitetty Isikdag et al. (2007) esittämä tiedonsiirtomenetelmien jako viiteen eri tapaan:

- (1) tiedostot
- (2) sovellusten yhteinen rajapinta
- (3) keskitetty palvelin
- (4) hajautettu palvelin
- (5) web-palvelu.

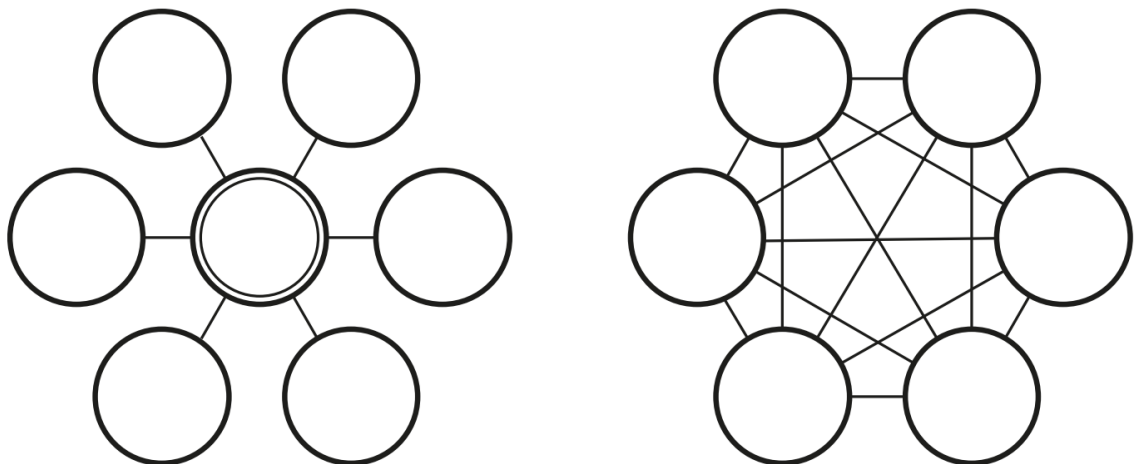
Olipa tiedonsiirron toteutus minkäläinen tahansa, edellyttää kahden järjestelmän välinen tiedonsiirto aina yhteistä rajapintaa, jossa määritellään, miten tiedonsiirto tapahtuu. Tässä diplomityössä keskitytään tiedostopohjaiseen tiedonsiirtoon yhteisen standardin avulla.



Kuva 14. Ylimpänä näkyy, että sovellukset kykenevät vaihtamaan tietoa, mikäli niillä on yhteinen tiedostomuoto, jota molemmat ymmärtävät. Toinen mahdollisuus on, että sovelluksilla on yhteinen rajapinta eli API (engl. application program interface). Sovellukset voivat myös kommunikoida yhteisen palvelimen kautta, tai siten, että sovellukset lukevat tietoa omalta palvelimeltaan ja palvelimet kykenevät kommunikoimaan keskenään. Alimpana on esitetty vaihtoehto, jossa sovellukset kommunikoivat web-api:n välityksellä.

Standardityyppejä on monia erilaisia, joista tietotekniikassa yleisiä ovat epäkaupallinen avoin standardi ja sen vastakohtana kaupallinen niin sanottu *de facto* -standardi. *De facto* -standardi on usein kaupallisen kilpailun seurauksena syntynyt yleisesti käytetty standardi, kuten Autodeskin kehittämä dwg-tiedostomuoto, joka 80- ja 90-luvuilla saavutti keskeisen aseman erityisesti rakennussuunnittelussa. Toinen yleinen standardityyppi on epäkaupallinen avoin standardi, josta esimerkkinä tietomallimuotoisen datan siirtämiseen kehitetty IFC-tiedonsiirtostandardi, jonka etuliitteen akronyymi tulee englannin kielen sanoista *industry foundation classes*.

Tietomallintamisen kehityksen alkuvaiheissa ei kehittynyt yhtenäistä tapaa siirtää tietomallimuotoista tietoa. Tämä on Howell & Batchelerin (2005) arvion mukaan aiheuttanut tietomallisovellusmarkkinoilla kilpailutilanteen, jossa ohjelmistovalmistajat pyrkivät saamaan taloudellista hyötyä omilla tiedonsiirtomuodoillaan. Kuvassa 15 on havainnollistettu Ibrahim et al. (2004) esittämiä tulevaisuuden kehityssuuntia tilanteelle. Joko ohjelmistovalmistajat pyrkivät laajentamaan toimialueitaan kattamaan mahdollisimman laajasti eri suunnittelualojen tarpeita, tai ohjelmista tehdään joustavia erikoistuneita järjestelmiä, jotka kykenevät kommunikoimaan yhteisen viitemallin avulla (Ibrahim et al. 2004). IFC-standardi edustaa edellä mainituista kehityssuunnista jälkimmäistä.



Kuva 15. Semanttinen esitys sovellusten kehitysmahdollisuuksista, missä vasemmalla sovellusten (ympyrät) välinen tiedonsiirto tapahtuu viitemallin avulla, kun taas oikealla jokainen sovellus on kehittänyt itsenäisen tavan siirtää tietoa muiden sovellusten kanssa. (Ojala 2013, 34.)

3.4.3 IFC-standardin visio, haasteet ja mahdollisuudet

IFC-standardin ydinajatuksena ja visiona on olla neutraali ja ohjelmistoriippumaton tiedonsiirtoformaatti, jonka avulla voidaan siirtää tietomallimuodossa olevaa tietoa eri järjestelmien ja ohjelmien välillä. IFC-standardia kehittää ja ylläpitää kansainvälinen organisaatio BuildingSMART. Sen visiona on kansainvälinen, avoin ja neutraali teknologia, joka mahdollistaa tehokkaan tiedonkulun koko rakennuksen elinkaaren ajan. Vision yksi kolmesta kulmakivistä on juuri IFC-standardi, joka määrittelee, miten tietomallimuotoista tietoa siirretään. Kaksi muuta ovat IFD- (engl. *international framework for dictionaries*) ja IDM-standardi (engl. *information delivery manual*). IFD määrittelee mitä tietoa jaetaan ja vaihdetaan, kun taas IMD määrittelee missä vaiheessa mitäkin tietoa jaetaan tai vaihdetaan. Näitä kahta viimeksi mainittua standardia ei käsitellä tarkemmin tämän diplomityön puitteissa.

Huolimatta siitä, että IFC-standardia on kehitetty noin 20 vuotta (1994 lähtien), se ei vielä kata kaikkia rakennuksen elinkaaren tiedonsiirtotarpeiden osa-alueita. Esimerkiksi geometrialtaan monimutkaiset rakenteet saattavat edelleen siirtyä virheellisesti eri sovellusten välillä (Jeong et al. 2009), tai niiden visualisointi saattaa olla virheellinen (Steel et al. 2010). Pazlar & Turk (2008) pitävät IFC-tiedonsiirtostandardia johtavana kandidaattina ratkaisemaan tietomallimuotoisen tiedonsiirron haasteet, mutta listaavat viisi syytä, jotka hidastavat kehitystä. Ne ovat:

- (1) rakennusalan pirstaloituneisuus
- (2) tuotteiden eli rakennusten ainutlaatuisuus
- (3) suunnittelualojen epäyhtenäiset yksityiskohtien tarkkuusvaatimustasot
- (4) takertuminen perinteisiin työmenetelmiin
- (5) vaatimustason jatkuva nosto.

IFC-tiedonsiirrossa on ollut myös tapauksia, joissa rakennusosan tyyppi tai sille määritetty materiaali eivät siirry johdonmukaisesti tietomalli- tai sitä hyödyntävien sovellusten välillä (Steele et al. 2010). Esimerkiksi korkea palkki saattaa muuttaa luokitustaan seinäksi, joka varsinkin rakennesuunnittelun kannalta tuottaa vaikeuksia laskennassa, koska reunaehdot palkin ja seinän tarkasteluissa ovat eriäviä. Samoin esimerkiksi teräsosissa on ollut tapauksia, joissa materiaali kirjautuu rakenneosan kuvaukseen, sen sijaan että sovellus ymmärtäisi sen tarkoittavan materiaalityyppiä (Steele et al. 2010).

Huolimatta ongelmista, on IFC-pohjainen tiedonsiirto Pazlarin & Turkin (2008) mukaan noussut erääksi varteenotettavimmaksi vaihtoehdoksi tietomallimuotoisen tiedon välittämiseksi ohjelmien välillä. Kiviniemi et al. (2008) arvion mukaan IFC-standardia voidaan pitää laadukkaana ja laajasti ohjelmistojen tukemana tiedonsiirtotapana. IFC-standardille tulee todennäköisesti käymään samoin kuin CAD-käsitteelle on käynyt, eli kun se alkaa toimia riittävän hyvin, yksityiskohdat väistyvät ohjelmistojen käyttäjien näkökulmasta vähitellen taka-alalle (Penttilä et al. 2006).

Useat merkittävät tietomallisovellukset, kuten Autodesk, Bentley, Graphisoft ja Nemetschek, tukevat IFC-muotoista tiedonsiirtoa (Steel et al. 2010). Laaja ohjelmistojen kirjo onkin yksi IFC:n vahvuuksista (Rammant & Adriaenssens 2008). Yksi sen merkittävimmistä eduista on laaja julkinen skeematietokanta, eli rakenteiden relaatioita ja merkityksiä kuvaava tietovarasto (Sanguinetti et al. 2012, 318). IFC on myös ainoa avoin standardi, joka kykenee siirtämään paitsi geometriatietoja, myös objektien rakenteen (topologian) sekä materiaali- ja käyttäytymisattribuutit (Jeong et al. 2009). Sitä voidaankin pitää perustana seuraavan sukupolven tiedonsiirrolle (Jeong et al. 2009).

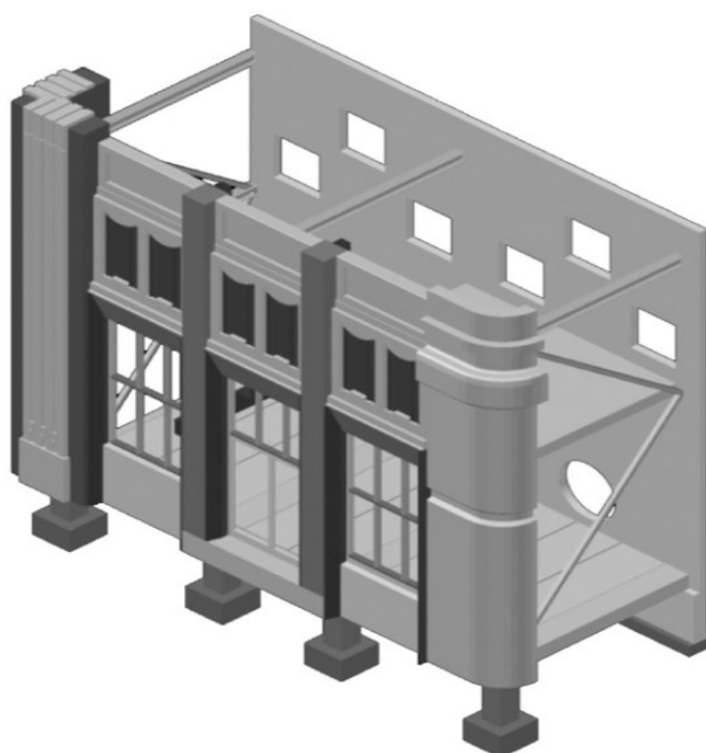
4 IFC-standardin avulla tapahtuvan tiedonsiirron testaus

4.1 Aiempia tutkimuksia ja tutkimusmenetelmän valinta

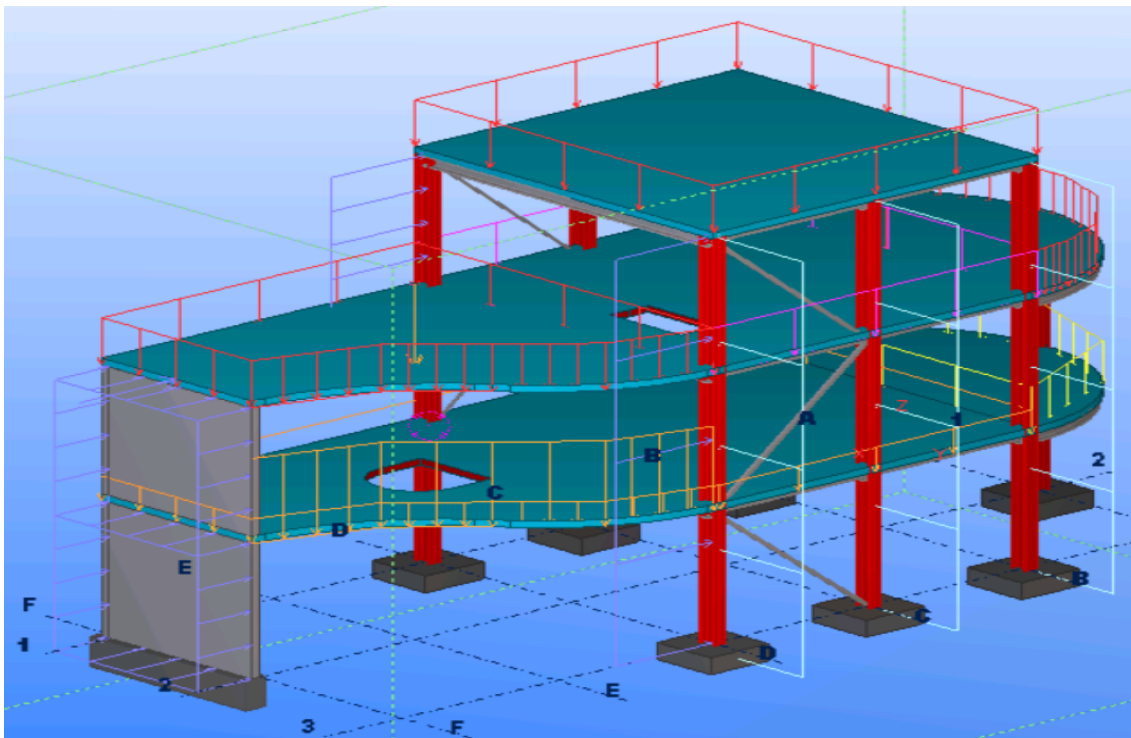
Eräs keino tietomallin ja laskentamallin välisen tiedonsiirron tutkimiseen on vertailutestimalli (engl. *benchmark*) tai lyhyemmin testimalli. Pitämällä testimallin sisältö samana, mutta siirtämällä mallin dataa eri tavoin ja eri asetuksilla, voidaan saada selville miten eri tiedonsiirtotavat eroavat toisistaan. Testimallin ei välttämättä tarvitse olla realistinen rakennuksen tietomalli, varsinkaan jos tarkoituksena on vain ja ainoastaan tutkia ohjelmien välisen rajapinnan kykyä välittää tietomallimuotoista dataa. Tutkittaessa yhteentoimivuutta ei pelkästään tarkastella kykyä välittää dataa, vaan myös sitä miten ohjelmat kykenevät rutiininomaisesti käyttämään tietoa omiin tarkoituksiinsa.

Yhteentoimivuutta testimallin avulla ovat tutkineet esimerkiksi Jeong et al. (2009), joiden käyttämä kuvitteellinen rakenne on esitetty kuvassa 16. Heidän testimallinsa on varsin realistinen rakenneosia, jonka avulla he tutkivat betonielementtijulkisivun suunnittelutiedon siirtymistä IFC-muodossa arkkitehtisuunnittelun käyttämän tietomalliohjelman ja elementtivalmistajan ohjelman välillä. Kuvassa 17 on Hejnfelt & Øksengaardin (2007) käyttämä testimalli, joka on selvästi rakennusteknisesti epärealistisempi. On kuitenkin otettava huomioon, että he tutkivat yleisellä tasolla eri tiedonsiirtotapojen ominaisuuksia tietomalliohjelmien välisessä tiedonsiirrossa. Siinä tarkoituksessa epärealistinen,

mutta useita erilaisia rakenteita ja materiaaleja sisältävä tietomalli palvelee paremmin tutkimuksen tavoitteita. Myös Pazlar & Turk (2008) hyödynsivät IFC-pohjaisen tiedonsiirron tutkimuksessaan testimallia, vaikka he lähestyivät aihetta ohjelmistotekniikan näkökulmasta. Tutkimuksessaan he vertailivat tietomalliohjelmista tuotettujen IFC-tiedostojen sisältöä ja niiden mahdollisia vääristymiä sekä muutoksia.



Kuva 16. Jeong et al. (2009) käyttämä testimalli, jonka avulla he selvittivät betonielementtijulkisivujen suunnittelutiedon siirtymistä IFC-muodossa tietomallisovelluksen ja elementtivalmistajan käyttämän sovelluksen välillä.



Kuva 17. Hejnfelt & Øksengaard (2007) testasivat kuvitteellisella rakennuksella erialisia tiedonsiirto-ominaisuuksia ja -reittejä tietomalli- ja rakenneanalyysisovellusten välillä. Rakennus ei ehkä toteuttaisi arkkitehtuurin tai lainsäädännön vaatimuksia, mutta käytetyt rakenteet ja liitokset ajavat asiansa tutkittaessa tiedonsiirtoa.

Edellä esitettyjen aiempien tutkimusmenetelmien perusteella voidaan todeta, että testimallia voidaan hyödyntää monenlaisissa eri tutkimustapauksissa ja eri lähestymistavoissa, kun halutaan selvittää tiedonsiirtoon liittyviä ongelmia. Kuitenkin testimallit sisältävät tyypillisesti toiminnoiltaan ja käyttäytymiseltään lähes identtisiä rakenneosia. Esimerkiksi Hejnfelt & Øksengaardin (2007) testimallissa pilareiksi luokiteltavia rakenneosia oli 19 kappaletta, joista useat olivat toistensa kopioita tai muodoltaan lähes yhteneviä. On kuitenkin mielekästä olettaa, että mittasuhteiden muuttaminen ei aiheuta tiedonsiirrossa ongelmia. Toisin sanoen, kyky siirtää dataa sovellusten välillä ei ole riippuvainen esimerkiksi pilarin koosta, vaan ennemminkin rakenneosien profiileista, muodoista ja asemoinnista. Oletus on looginen, koska tällöin muuttuu myös rakenteen toiminta. Tästä syystä kokonaisen testimallin sijaan on katsottu mielekkäämmäksi siirtää yksittäisiä rakenneosia, joiden geometria poikkeaa toisistaan muilta osin kuin vain mittasuhteilta. Esimerkiksi siirtämällä poikkileikkaukseltaan muuttuvia tai profiililtaan erilaisia rakenneosia voidaan paremmin saada selville tiedonsiirron ongelmat ja rajoitteet.

Kirjallisuuden perusteella edellä esitettyjen tutkimusten kaltaisia tarkasteluita tietomallimuotoisen tiedon siirrolle on olemassa varsin runsaasti. Tyypillisesti niiden pääpaino on kuitenkin vertailla erilaisia tiedonsiirtomenetelmiä keskenään, ja tuoda esiin menetelmien etuja ja haittapuolia. Ne tutkimukset, joissa aiheena on IFC-muotoinen tiedonsiirto, keskittyvät usein sen tieto- tai ohjelmistotekniseen puoleen. Toisin sanoen, niissä tarkastellaan miten IFC-standardin lähdekoodi esittää ja käsittelee rakennuksen tietomallia. Nämä tutkimukset ovat varsin kaukana IFC:n keskivertokäyttäjistä. Tämän vuoksi tässä diplomityössä aihetta lähestytään käytännönläheisemmältä suunnalta tarkastelemalla tiedonsiirtoprosessia käyttäjän lähtökohdista, ja pohtimalla voiko käyttäjä vaikuttaa IFC-pohjaisen tiedonsiirtoprosessin sujuvuuteen.

4.2 Tutkimusasetelma, tavoitteet ja oletukset

Tässä tutkielmassa tietomallimuotoisen tiedonsiirron tarkastelun kohteeksi on valittu kolme rakennesuunnittelun kannalta merkittävää asiasisältöä. Ensimmäinen ja ehkä merkittävin on geometria, jonka sujuvalla siirtymisellä vältetään laskentamallissa uudelleenmallintamiselta, joka tyypillisesti on mitoitusprosessin työläin ja eniten aikaa vievin vaihe. Materiaali- ja kuormitustietojen siirtyminen osaltaan nopeuttaa laskentaprosessia. Kuormitustiedot määritellään tyypillisesti vasta laskentamallissa, mutta niiden määrittely jo tietomallinnusvaiheessa vähentäisi merkittävästi laskentamallissa tapahtuvaa mallinnusta. Kaikkien näiden kolmen siirtyminen sujuvasti ja virheettömästi vähentäisivät oleellisesti inhimillisen virheen riskiä ja tehostaisivat mitoitusprosessia.

Koejärjestelyssä testattaviksi rakenneosiksi on valittu TS-tietomalliohjelman perustyökaluilla mallinnettavat osat, eli pilari, palkki, seinä ja laatta. Jokaista rakenneosatyyppiä mallinnetaan kahdeksan kappaletta eri tavoin ja eri profilein. Liittämällä rakenneosiin lisäksi erilaisia materiaalitietoja sekä kuormituksia saadaan selville myös sovellusten kyky siirtää materiaaliarvoja sekä kuormituksia IFC-muodossa.

Koejärjestelyn tavoitteena on kartoittaa miltä osin rakenteita voidaan siirtää sujuvasti tieto- ja laskentamallien välillä. Samalla tavoitteena on selvittää ja tunnistaa ne rakenteet, joiden kohdalla ongelmia saattaa tiedonsiirtoprosessissa ilmetä. Lisäksi tavoitteena on tutkia, voidaanko materiaali- ja kuormitustietoja siirtää luotettavasti IFC-muotoisessa tiedonsiirrosta. Koejärjestelyitä ja tiedonsiirtoprosessia tarkastellaan ja arvioidaan myös käyttäjän näkökulmasta. Käyttäjäksi on tässä diplomityössä määritetty rakennesuunnittelija, jolla on ymmärtämys rakenteiden toiminnasta, mutta jolla ei ole syvällisempää tietämystä ohjelmistotekniikasta tai IFC-muotoisesta tiedonsiirtoprosessista. Näin ollen käyttäjän kannalta keskiössä on tiedonsiirtoprosessin vaivattomuus, mukautuvuus sekä opettavuus.

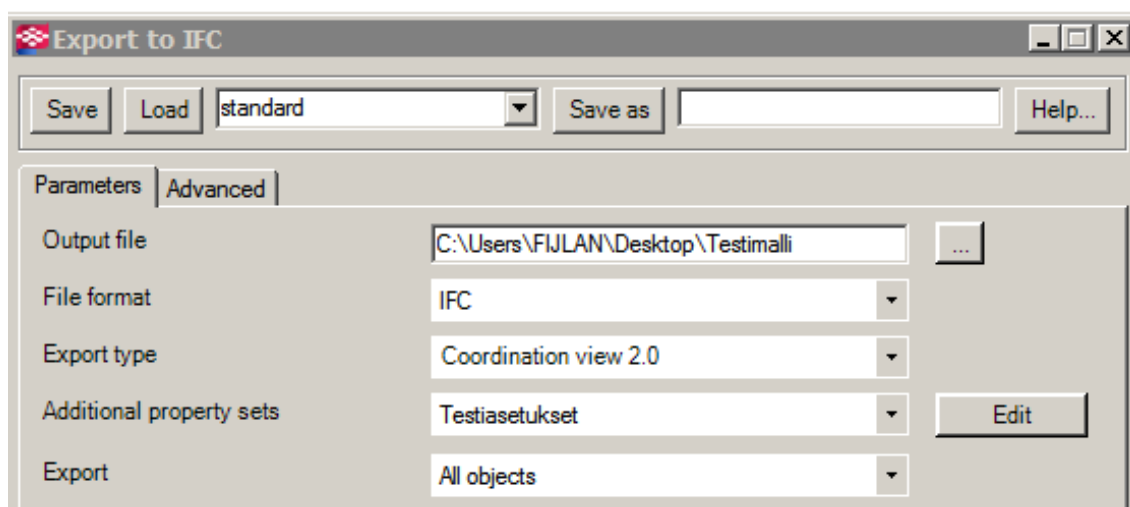
Testauksessa käytettiin tietomalliohjelmistona Tekla Structures -ohjelman versionumeroa 20.1. Ohjelmaan viitataan jatkossa nimityksellä tekla tai lyhenteellä TS. Ohjelmiston eri päivitysversionot ovat laajasti käytössä suomalaisissa rakennesuunnittelutoimistoissa. Laskentamalliohjelmana käytettiin Robot Structural Analysis Professional -ohjelman versionumeroa 2015. Ohjelmaan viitataan jatkossa nimityksellä robot tai lyhenteellä RSA. Teklaa kehittää Trimble, joka on monikansallinen ja monialainen paikkatietopohjaiseen teknologiaan keskittynyt yritys. Robottia kehittää Autodesk, joka on monikansallinen CAD-ohjelmistoihin erikoistunut yritys. Molemmat ohjelmat ovat alallaan menestyneitä ja tunnettuja, jonka vuoksi voidaan olettaa, että suoritettavat kokeet voidaan riittävällä tarkkuudella toistaa millä tahansa vastaavalla kilpailukykyisellä ohjelmalla.

Tieto- ja laskentamallien käyttöön liittyy olennaisesti myös inhimillinen toiminta. Tietoteknisten järjestelmien nopea kehitys vaatii toimijoilta kehitysmuutoksista asennetta ja organisaatioiden toimintatapojen muutoksia. Kehityksen ja muutosten aiheuttama inhimillinen kitka jätetään kuitenkin tarkastelun ulkopuolelle, ja oletetaan toimijoiden olevan kehitysmuutoksia ja kyvykkäitä sopeutumaan muuttuviin järjestelyihin. Aihetta tarkastellaan oletuksella, että kaikki ovat halukkaita sitoutumaan.

4.3 Geometriatiedon siirtäminen IFC-tiedostoksi

4.3.1 Ohjelmien asetukset

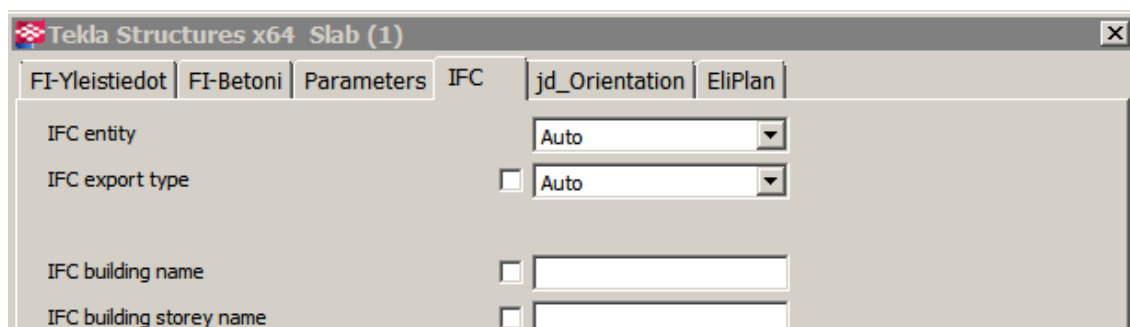
Siirrettäessä tietomallia TS-ohjelman sisäisestä muodosta IFC-muotoon, merkittävimmät käyttäjän tekemät valinnat ovat tiedostomuoto (engl. *file format*) ja vientimuoto (engl. *export type*). Kuvassa 18 on esitetty TS-ohjelman käyttämä asetusvalikko, joka avautuu, kun käyttäjä haluaa muuttaa tietomallin tietoa IFC-muotoon. Tiedostomuodossa ilmaistaan talletetun tiedoston rakenne, kun taas vientimuodossa määritellään tiedoston rakenteen skeema eli kuinka rakenne muodostetaan. Tiedostomuotona testaamisessa on käytetty IFC:tä, eli alkuperäistä pakkaamatonta tiedostomuotoa IFC-standardin määrittelemää rakennetta. Vientimuotona on käytetty coordination view 2.0, joka on buildingSMART uusin IFC-skeeman tiedostorakenne.



Kuva 18. TS-ohjelmassa avautuva valikko, kun käyttäjä haluaa luoda IFC-tiedoston. Parametrit-välilehdellä voidaan muuttaa tiedostonluonnin asetuksia.

Lisäksi kuvassa 18 voidaan erottaa asetusten kohta lisäominaisuusjoukot (engl. *additional property sets*), johon on määritetty tätä testiä varten oma asetus nimeltä testiasetukset. Käyttäjä voi halutessaan määrittää lisäominaisuusjoukko-asetusten avulla mitä kaikkia IFC-entiteettejä (engl. IFC-entity) ryhmiä ja näiden sisältämiä attribuutteja käytetään hyödyksi IFC-tiedoston muodostamisessa. Valittavana on monia satoja eri vaihtoehtoja, eikä niitä käydä tässä tutkimuksessa tarkemmin läpi. Useimpien attribuuttien sisällön pystyy päättämään nimestä, kuten esimerkiksi IFCBeam-entiteetin weight-attribuutti kertoo palkin painon. Tätä testiä varten luodussa asetuksessa nimeltä testiasetukset, valittiin kaikki mahdolliset attribuutit.

Näiden asetusten lisäksi käyttäjä voi yksittäisten rakenneosien kohdalla valita IFC-entiteetin sekä kappaleen esitystavan IFC-tiedostorakenteessa. Kuvassa 19 on esitetty TS-ohjelman valikko, josta yksittäisten rakenneosien käyttäytymistä voidaan muuttaa. Auto-valinnalla TS-ohjelma määrittää automaattisesti sopivimmaksi katsomansa entiteettityypin rakenteelle, mutta käyttäjä voi halutessaan muuttaa tätä. Esimerkiksi palkkityökalulla mallinnetun korkean seinän kohdalla voi olla mielekästä vaihtaa entiteetiksi IFCWall. Kappaleen esitystavassa TS-ohjelma ensisijaisesti pyrkii esittämään kappaleen swept-solid-geometriaa, joka tarkoittaa, että profiilit ja alueet määritetään oman akselinsa suhteen pursotettuina kappaleina. Brep-vaihtoehdossa taas kappaleet määritellään kärkipisteidensä avulla siten, että ne muodostavat tilavuuskappaleita.

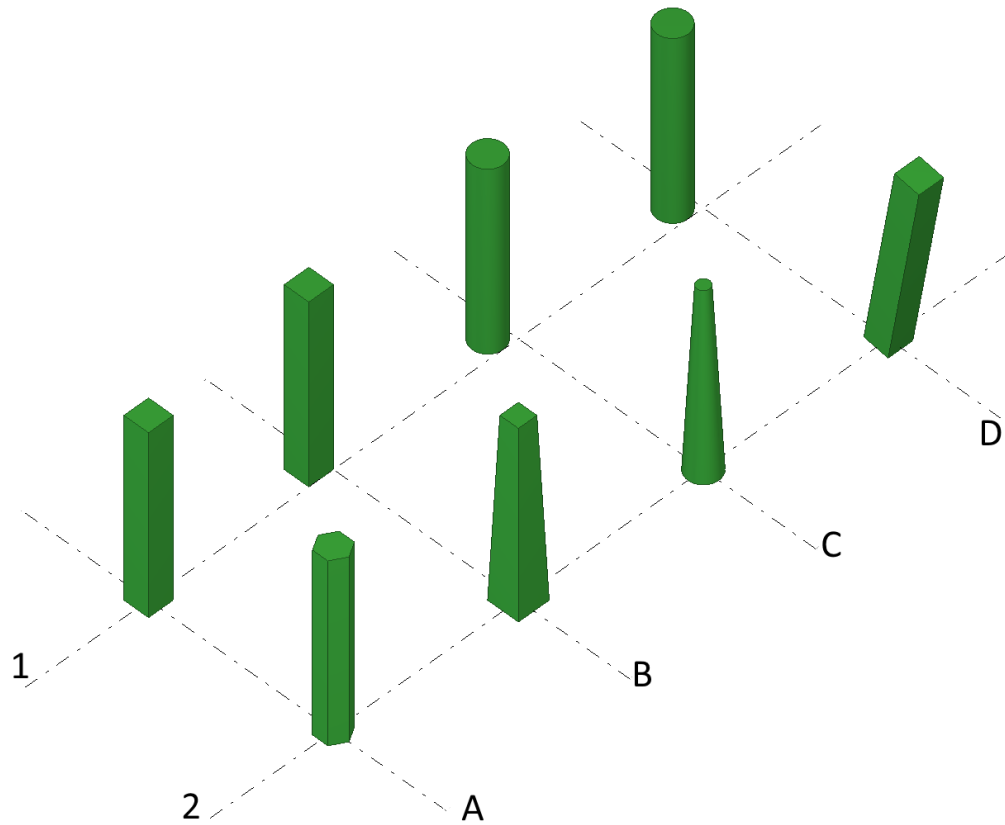


Kuva 19. TS-ohjelman kuvakaappauksessa näkyy erään laatan IFC-asetukset. Kaikissa testeissä kappaleen asetuksina molemmissa kohdissa oli auto, jolloin ohjelma pyrkii valitsemaan automaattisesti sopivimman asetuksen kappaleelle.

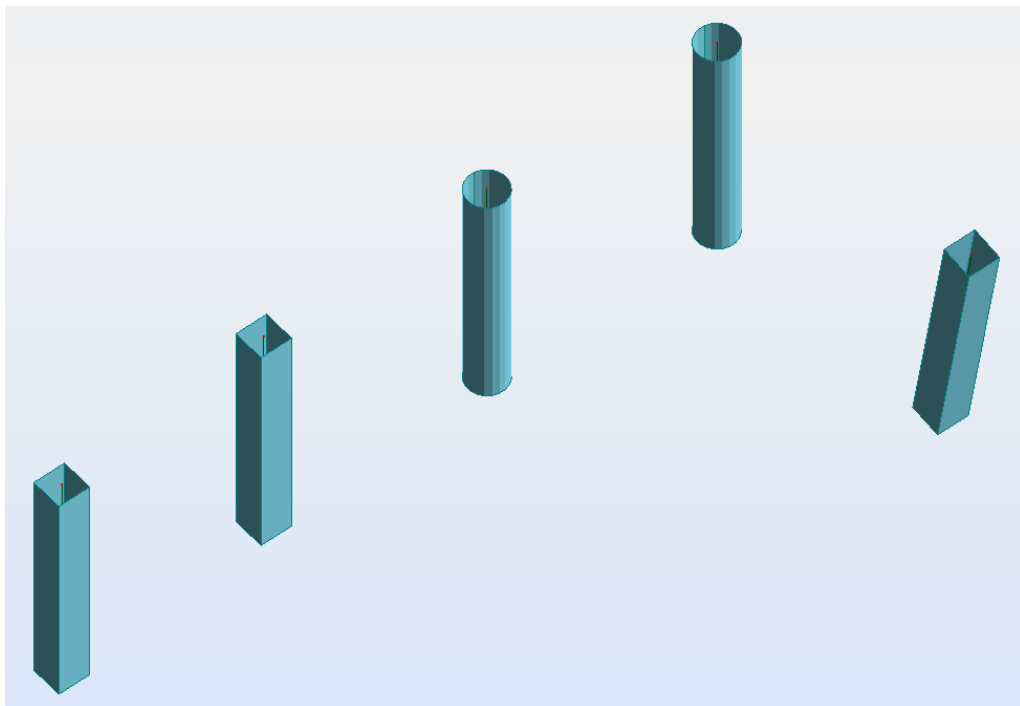
RSA-ohjelmalla avattaessa IFC-tiedostoja, ei ole erillistä asetusvalikkoa jolla käyttäjä voisi vaikuttaa siirtyvään dataan. Ohjelmalla IFC-tiedosto avataan samoin, kuin RSA-ohjelman sisäisen muodon tiedostot. Tiedoston avauduttua ohjelma esittää kappaleet omalla grafiikallaan, joka luonnollisesti poikkeaa TS-ohjelman grafiikasta. Mikäli RSA-ohjelma ei kykene tunnistamaan rakenneosaa tai sen profiilia, se pyrkii esittämään kappaleen geometrian pintoina. Esimerkiksi jos RSA ei kykene tunnistamaan palkkia, esittää se palkin kuutena suorakaidepintana, jotka vastaavat alkuperäistä geometriaa.

4.3.2 Geometriatiedon siirtyminen kuvin esitettynä

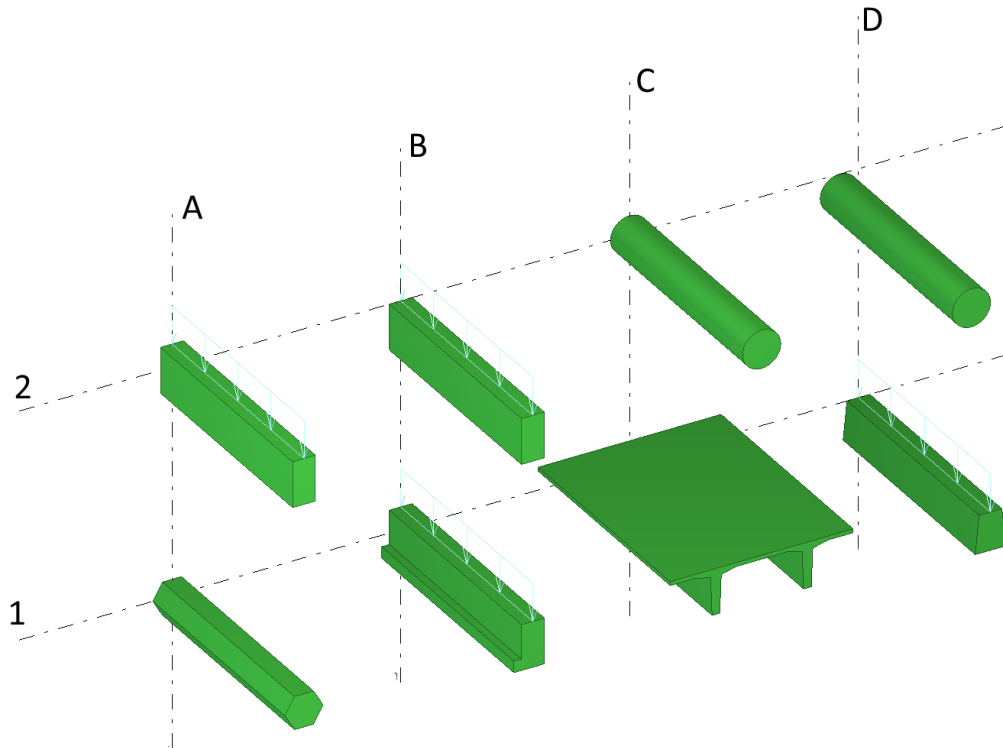
Tässä kappaleessa esitetään kuvapareina testattujen yksittäisten rakenneosien siirtyminen. Kuvaparin ensimmäisessä kuvassa esitetään kappaleet niin kuin ne oli mallinnettu TS-ohjelmassa. Mallintamisessa oli käytetty rakenneosan omaa betoniosatyökalua, eli pilari-, palkki-, seinä- tai laattatyökalua. Rakenneosat sijaitsivat moduulilinjastolla, jonka linjojen etäisyys toisistaan oli kolme metriä. Rakenneosien pituus oli niin ikään kolme metriä. Osien poikkileikkaukset ja rakennepaksuudet vaihtelivat 300-600 mm välillä. Kuitenkin kuten aiemmin todettiin, rakenneosien dimensiot eivät testin lopputuloksen kannalta ole merkityksellisiä. Kaikkien rakenneosien materiaaliksi asetettiin betoni C30/37. Kuvaparien alemmassa kuvassa on esitetty RSA-ohjelman näkymä sellaisena, kuin ohjelma oli rakenneosien datan IFC-tiedostosta ohjelmaan tuonut.



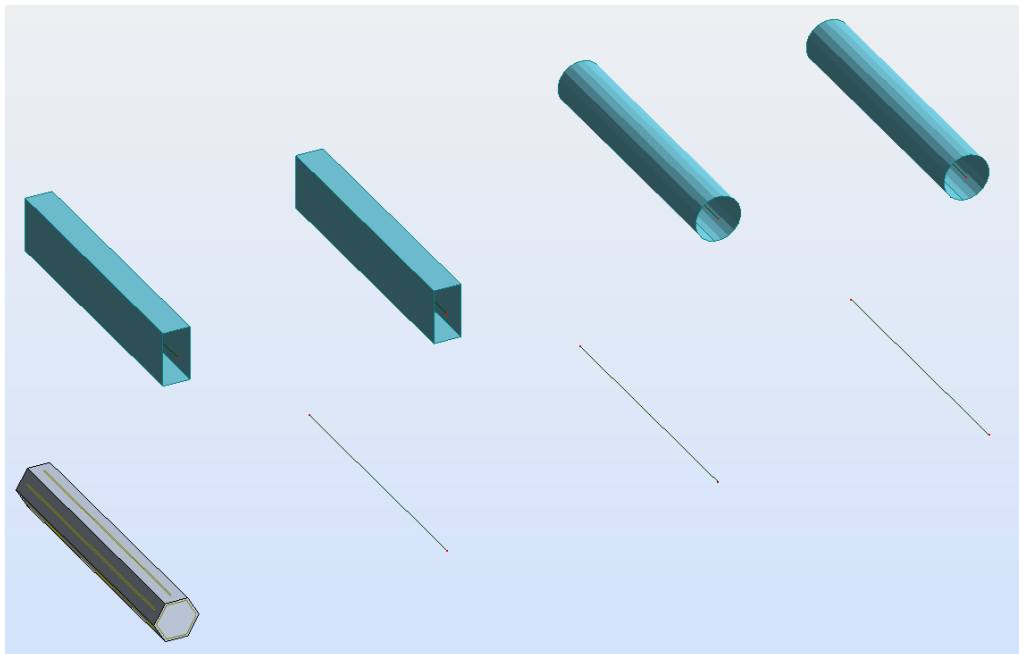
Kuva 20. TS-ohjelmassa mallinnettiin kahdeksan pilaria. Linjalla yksi on tyypillisiä neliön ja pyöreän muotoisia betonipilareita. Linjalla 2 puolestaan profiloituaan erikoisempia muotoja; paitsi D2 pilari, joka on vino.



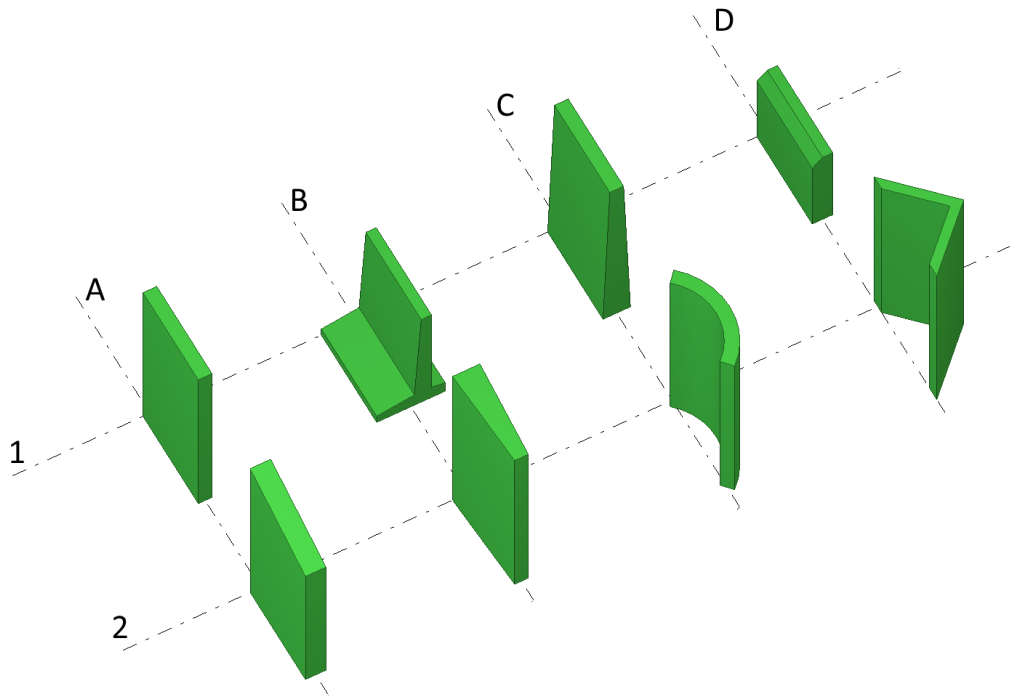
Kuva 21. IFC-standardin avulla RSA-ohjelmaan siirtyneet pilarit, joissa ohjelma tunnisti myös analysiiviat. Ulkonäöstään huolimatta pilarit eivät ole onttoja, vaan kysymys on RSA-ohjelman esitystavasta. Pilarit A2, B2 ja C2 eivät siirtyneet ohjelmaan.



Kuva 22. TS-ohjelmassa mallinnetut kahdeksan palkkia. Pilarit 2A ja 2B ovat tavallisia betonipalkkeja, poikkileikkaukseltaan suorakaiteita. Linjalla 1B on leukapalkki ja linjalla 1C TT-palkki. Muut palkeista ovat profiililtaan erikoisempia. Lisäksi palkeissa 1B, 1D 2A sekä 2B on asetettu vaikuttamaan viivakuorma.



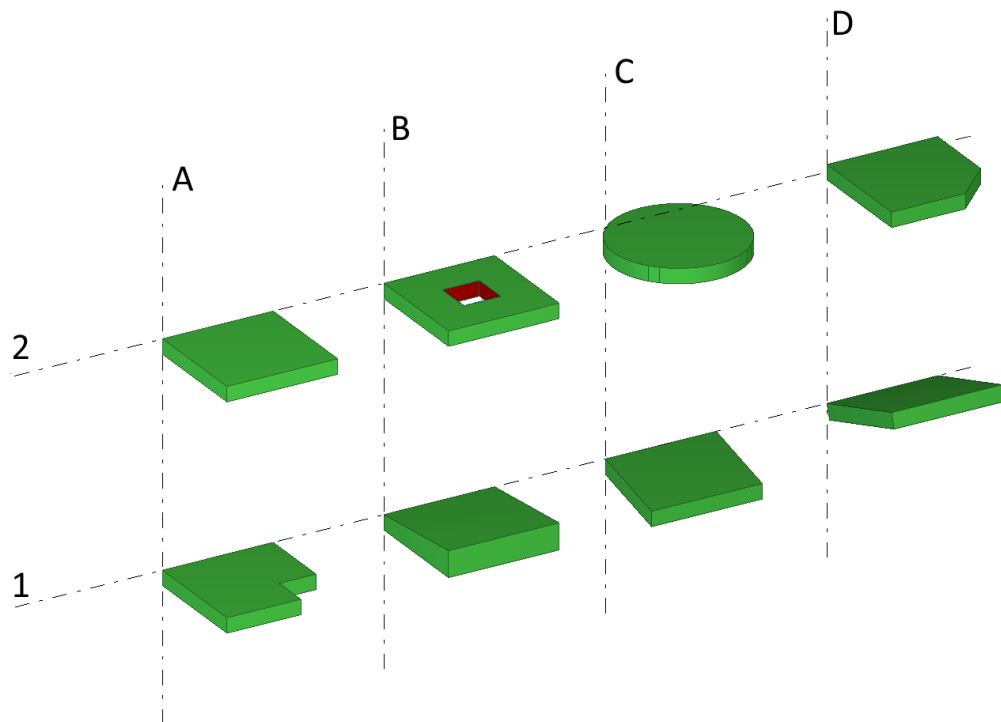
Kuva 23. IFC-standardin avulla RSA-ohjelmaan siirtyneet palkit, joista ohjelma kykeni tunnistamaan muiden palkkien analyysiviivan, paitsi profiililtaan kuusikulmaisen palkin, joka siirtyi pintoina. Sen sijaan palkkien 2B, 2C ja 2D profiileita ohjelma ei kyennyt tunnistamaan. Kuormitustietoja ohjelma ei tunnistanut. Ulkonäöstään huolimatta ylärivin palkit eivät ole onttoja, vaan kysymys on RSA-ohjelman esitystavasta.



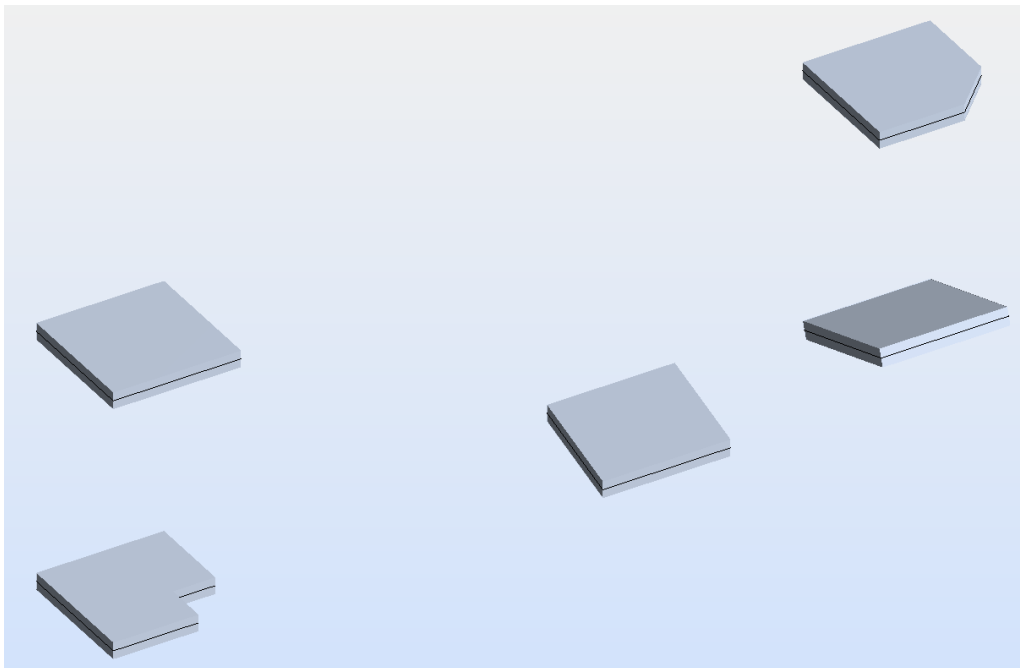
Kuva 24. TS-ohjelmassa mallinnetut kahdeksan seinää. Mielikuvituksellisempia seiniä ovat 2C ja 2D linjojen kaareva ja 90-asteen kulman muodostava seinä. palkkia. Linjan 2D seinässä ei siis ole kyse kahdesta erillisestä seinästä.



Kuva 25. IFC:n avulla RSA-ohjelmaan siirretyt seinät, joista vain yhden profiilin ohjelma kykeni tunnistamaan. Seinät linjoilla 2C ja 2D eivät siirtyneet lainkaan.



Kuva 26. TS-ohjelmassa mallinnetut laatat, joihin lisättiin myös neliön muotoiset reikä laattaan 2B.



Kuva 27. IFC-standardin avulla moni laatoista siirtyi oikein. Vain laatta jossa oli reikä (2B), pyöreä laatta (2C) ja paksuneva laatta (1C) eivät siirtyneet lainkaan.

5 Pohdinta

5.1 Havainnot tuloksista ja tutkimusprosessista

Koetulosten perusteella voidaan yleisellä tasolla todeta, että rakennesuunnittelussa tarvittava tiedon siirtäminen tietomallista laskentamalliin IFC-standardiin pohjautuvan tiedonsiirron avulla on rajoitetusti mahdollista. Geometrialtaan yksinkertaiset rakenteet, kuten esimerkiksi suorat palkit tai pilarit, on mahdollista siirtää ohjelmien välillä. Sen sijaan geometrialtaan monimutkaisemmat rakenteet, kuten kaarevat tai kapenevat osat aiheuttavat merkittäviä ongelmia. Lisäksi testauksessa materiaali- tai kuormitustietoja ei saatu siirrettyä lainkaan.

TS-ohjelmassa valitun mallinnuspisteiden valinnalla ei näyttänyt olevan mitään vaikutusta rakenneosien analyysiviivaan, joka siirretyissä rakenneosissa näytti kulkevan aina rakenneosan geometrian keskellä. Epäselväksi jäi miksi RSA-ohjelma kykeni tunnistamaan rakenneosan analyysiviivan, mutta ei itse profiilia. Esimerkiksi seinien kohdalla tällaisia tapauksia oli useita.

Kiinnostava havainto palkkien ja pilarien siirrossa oli kuusikulmainen profiili. Pilareiden siirrossa RSA-ohjelma ei lukenut kuusikulmaista profiilia lainkaan, kun taas palkkien siirrossa se siirtyi pintoina. Kyse on kuitenkin tismalleen samanlaisesta rakenneosasta, vaikka toinen on pystysuorassa ja toinen vaakatasossa. Eron aiheutunee kenties siitä, että osat ovat TS-ohjelmassa mallinnettu eri työkalulla tai siitä, miten IFC-entiteetti kuvaa rakenneosan IFC-tiedostoon.

Huomionarvoinen havainto oli myös laattojen 2A ja 2B siirtymisessä. Kyseiset laatat olivat geometrialtaan identtisiä, mutta 2B-laatasta oli neliön muotoinen reikä keskellä laatataa. Siinä missä 2A-laatta siirtyi RSA-ohjelmaan, ei 2B-laatta siirtynyt lainkaan. TS-ohjelmassa reikä oli tehty laattaan leikkaustyökalulla, jonka avulla määritettiin mitä rakenneosaa leikataan, mistä kohtaa ja kuinka syvältä. Nähtävästi reikätyökalulla on merkitystä myös siihen, miten itse laatta IFC-tiedonsiirron avulla siirtyy.

Tarkasteltaessa tiedonsiirtoprosessia käyttäjän näkökulmasta, voidaan merkittävänä hankaluutena pitää itse prosessin monimutkaisuutta. Ohjelmissa on useita eri vaihtoehtoja asetuksille, ja näiden tarkka tunteminen vaatii pitkällistä perehtymistä ohjelman toimintaan ja laajaa käytännön kokemusta tiedonsiirrosta. Lisäksi tutkimuksen aikana kävi ilmi, että ohjelmistovalmistajien tuottamien ohjeiden tietosisältö oli usein pinnallista tai puutteellisia. Esimerkiksi IFC-standardiin perehtymättömälle henkilölle pelkästään ohjeissa esiintyvien termien ymmärtäminen on haasteellista. Tutkimusprosessissa parhaimpana oppimismenetelmänä vaikutti olevan erilaiset internetin keskustelufoorumit, joissa tieto- ja laskentamallien käyttäjät esittävät omia ongelmiaan ja kyselevät toisilta käyttäjiltä ratkaisuja näihin.

5.2 IFC-pohjaisen tiedonsiirron hyödyntäminen

Tulosten perusteella näyttää siltä, että IFC-pohjainen tiedonsiirto sopii nykyhetkellä parhaiten kohteisiin, joissa rakenteet ovat yksinkertaisia ja toistavat itseään. Esimerkkinä tällaisesta kohteesta ovat erilaiset muodoltaan yksinkertaiset varasto- ja tuotantorakennukset. Tulevaisuudessa tiedonsiirron kykenevyyden kehittyessä IFC-pohjaista tiedonsiirtoa voidaan hyödyntää myös monimutkaisemmissa rakennuksissa, kuten esimerkiksi geometrialtaan ainutlaatuiset julkiset rakennukset. Soveltaminen nykyisellään ja lähitulevaisuudessa edellyttää, että rakenteiden ongelmakohdat kyetään tunnistamaan aikaisessa vaiheessa projektia, ja että IFC-pohjaista tiedonsiirtoa sovelletaan ensisijaisesti helpoksi havaittuihin osiin.

On kiistatta selvää, että IFC-pohjaisella tiedonsiirrolla ei pystytä vastaamaan tieto- ja laskentamallin välisiin tiedonsiirron haasteisiin betonirakenteiden rakennesuunnittelussa. Kun suunnittelu etenee ja rakenteet tarkentuvat, saavutetaan ennen pitkään tilanne, jossa rakenteiden siirtämisellä laskentamalliin ja niiden muokkaaminen kelvollisiksi ei saavuteta taloudellista hyötyä rakennesuunnittelulle. Toisin sanoen on työläämpää siirtää ja muokata rakenteita, kuin esimerkiksi muokata vanhaa laskentamallia muuttamalla rakenteet uusimpien suunnitelmien mukaisiksi. Luonnollisesti tämä käsin tapahtuva muokkaaminen lisää inhimillisten virheiden riskiä heikentäen suunnitelmien laatua. Koko suunnitteluprosessin ohjauksen kannalta tämä tarkoittaa taloudellista vertailua laskentamallista saatavan hyödyn ja mallin korjailuun käytettyjen työtuntien välillä.

Voidaan myös todeta, mitä yksityiskohtaisemmaksi ja viimeistellymmäksi rakenteet muuttuvat, sitä enemmän tällainen simulointipohjainen rakennesuunnittelu, joka hyödyntää IFC-pohjaista tiedonsiirtoa tieto- ja laskentamallien välillä, vie aikaa ja resursseja. Kuitenkin kuten aiemmin todettiin, taloudellisesti merkittävämät rakenteelliset päätökset tehdään suunnitteluvaiheen alussa, jolloin myös yksityiskohtien määrä on verrattain vähäinen. Luonnossuunnittelun kannalta tämä tarkoittaa, että karkeasti mallinnettuja yksinkertaisia rakenteita on mahdollista jo aikaisessa vaiheessa suunnitteluprosessia siirtää tieto- ja laskentamallien välissä, jolloin tietoa rakenteiden toiminnasta voidaan saada lyhyellä aikavälillä. Näin rakennevaihtoehtoja voidaan kohtuullisen helposti vertailla keskenään ja päästä laadukkaampaan lopputulokseen. Edellä esitetyt arvelut perustuvat ajatukseen, että IFC-pohjaisella tiedonsiirrolla pyritään siirtämään kokonaisia rakennuksia tai suuria rakennuksen lohkoja laskentamalliin kokonaisstabiliteettilaskentaa varten.

Sen sijaan, jos tarkastellaan yksittäisten rakenneosan tai pienten rakenneosakokonaisuuksien siirtämistä, saattaa IFC-muotoinen tiedonsiirto tarjota potentiaalisen vaihtoehdon tiedonsiirtotavaksi. Siirtoprosessin etuna on, että yksinkertaisen rakenneosan geometria saadaan siirrettyä oikein tai lähes oikein laskentamalliin. Pienten muokkausten tekeminen laskentamallissa on pienellä vaivalla mahdollista, mutta suurempien kokonaisuuksien tarkastaminen ja muokkaaminen vaatii aikaa ja resursseja.

5.3 Tiedonsiirto tieto- ja laskentamallien välillä nyt ja tulevaisuudessa

Verrattaessa koejärjestelyistä saatuja käytännön kokemuksia kirjallisuudessa esitettyyn tietomallimuotoisen tiedonsiirron potentiaaliin, voidaan todeta tieto- ja laskentamalliohjelmien yhteentoimivuuden olevan vasta alkutekijöissään. Pelkästään tietomallilla tuotetun geometriatiedon siirto IFC:n avulla laskentamalliin vaatii molempien ohjelmistojen hyvää tuntemusta, jotta ymmärretään erilaisten asetusten vaikutukset juuri kyseisen ohjelmistoparin yhteentoimivuuden optimoimiseksi. Tämä on kaukana siitä kirjallisuudessa esitetystä potentiaalista, että laskentamalliin voitaisiin siirtää tietoa suoraan tietomallista – puhumattakaan siitä, että siirrettyä tietoa voitaisiin suoraan hyödyntää laskentamallissa. Toki pitää muistaa, että laadukas tieteellinen tutkimus esittää alansa huippuosaajien ja huipputeknologioiden saavuttamia tuloksia, eivätkä tässä mielessä vastaa keskivertokäyttäjän kokemusta asiasta.

Aiemmin tässä diplomityössä esiteltiin kuvassa 13 yhteentoimivuuden eri tasot. Kuvan luokitusta käyttäen voidaan arvioida, että IFC-muotoista tiedonsiirtoa käyttäen ollaan yhteentoimivuudessa lähellä tasoa yksi, eli tekninen yhteentoimivuus. Toisin sanoen eri järjestelmät kykenevät IFC:n avulla jossain määrin viestiä keskenään, mutta rutiininomainen toisen ohjelman tietojen käyttö on vielä kaukana. Käyttäjän näkökulmasta ajateltuna, IFC-muotoinen tiedonsiirto on monimutkaista, hankalaa ja epäluotettavaa. Käyttäjä pystyy hyvin vähän vaikuttamaan tiedonsiirron onnistumiseen, vaan se joko toimii tai ei toimi. Käytännössä ainoa mitä käyttäjä voi tehdä, on opetella tuntemaan ohjelmien asetukset, jolloin hän pystyy arvioimaan, kannattaako IFC-muotoista tiedonsiirtoa edes käyttää kyseisessä tapauksessa.

Tutkimusprosessin aikana tuli selvästi ilmi valtava potentiaali, joka onnistuneella tiedonsiirtoprosessilla ja yhteentoimivuudella on saavutettavissa. Rakennesuunnittelun kannalta tavoitteena on, että tietomallista siirrettävää dataa voidaan hyödyntää laskentamallissa ilman, että sitä joudutaan millään tavoin muokkaamaan tai käsittelemään. Ideaalitilanteessa laskentamallilla tuotettua dataa voidaan siirtää takaisin tietomalliin käsiteltäväksi ja hyödynnettäväksi. Tällaisen prosessin taloudellinen hyöty on ilmeinen, minkä lisäksi välttyään ihmisen tekemältä, inhimillisille virheille alttiilta työltä.

IFC-pohjainen tiedonsiirto on yksi mahdollisista keinoista saavuttaa kitkaton yhteentoimivuus eri suunnitteluohjelmistojen välillä. Sen etuna on avoimuus, mutta ennen kaikkea IFC kilpailee monipuolisella kattavuudellaan. Ohjelmistovalmistajat suosivat kaupallisen edun nimissä omia tiedonsiirtoformaattejaan. Koska kehitys vaatii aina resursseja, jää IFC tällaisessa kilpailuasetelmassa väistämättä toiseksi. Sen sijaan, mikäli se kykenee tarjoamaan laaja-alaista tiedonsiirtoa monien eri ohjelmien välillä, on sillä mahdollisuudet tulevaisuudessa saavuttaa hallitseva asema tietomallimuotoisessa tiedonsiirrossa.

Rakennusala on monissa yhteyksissä todettu olevan otollinen kehittyneen tietomalli- ja tiedonsiirtoteknologian käyttöönottoon, mutta toistaiseksi näin ei ole tapahtunut. Rakennesuunnittelussa toimivien yritysten etuna on, että ne ymmärtävät omat tarpeensa ja uuden teknologian tuomat mahdollisuudet. Yrityksen etu on, jos se kykenee toimimaan yhteistyössä ohjelmistovalmistajien kanssa, jolloin se pystyy mahdollisesti vaikuttamaan omalta kannaltaan edullisesti ohjelmistojen kehitykseen.

5.4 Tutkielman kriittinen tarkastelu, haasteet ja jatkotutkimus

Tämän diplomityön aiheen määrittelyn merkittävin haaste on poikkitieteellisyys. Rakennesuunnittelun kannalta tieto- ja laskentamallit ovat työkaluja, joilla työtetään ja esitetään suunnitelmia. Käytännössä rakennesuunnittelijalta voidaan vaatia käytössä olevien ohjelmien teknistä osaamista sillä tasolla, että hän kykenee tehokkaasti käyttämään niitä suunnittelun tukena. Tietotekniikan osaamisen merkitys on kiistatta korostunut nykyaikaisessa rakennesuunnittelussa, mutta rakennesuunnittelijan päätehtävä on edelleen rakennustekniikan ymmärtäminen ja rakenteiden kantavuuden osoittaminen lujuuslaskelmin. Tieto- ja laskentamallien yhteentoimivuuden sujuvoittaminen on tässä mielessä rakennesuunnittelijan ulottumattomissa.

Rakennesuunnittelijalla on käyttäjän näkökulmasta hyvin vähän keinoja vaikuttaa tieto- ja laskentamallien väliseen IFC-muotoiseen tiedonsiirtoprosessiin. Käytännössä ainoa mitä suunnittelija voi tehdä, on opetella tuntemaan käytössä olevien ohjelmien eri asetusten vaikutukset tiedonsiirtoprosessissa. Jotta aiheeseen ja ongelmiin päästäisiin pureutumaan syvällisemmin, tarvittaisiin monialaosaaja, jolla olisi ymmärtämystä rakennetekniikasta, tietotekniikasta ja ohjelmistotekniikasta. IFC-standardin lähdekoodi on avoin, joten teoriassa sitä voisi kehittää myös yksittäinen henkilö. Ongelmana kuitenkin on, että myös kaupallisten ohjelmien pitäisi kyetä sekä tuottamaan että lukemaan IFC-muotoista tietoa.

Toinen merkittävät ongelma on aiheen laava rajaus. Aihetta on lähestytty betonirakenteiden rakennesuunnittelun kulmasta, joka kuitenkin on näkökulmana laaja ja jota voi lähestyä monesta suunnasta riippuen tarkasteleeko asiaa suunnittelija, mallintaja tai projektinjohtaja. Esimerkiksi rakennesuunnittelijalle merkittävin asia voi olla se, miten sujuvasti tietomallin dataa voidaan hyödyntää laskentamallissa. Projektinjohtamisen kannalta kiinnostavampaa voi olla se, kuinka paljon työtunteja käytetään IFC-muotoisen tiedonsiirron ongelmien selvittelyyn, ja missä vaiheessa on taloudellisempi ratkaisu kopioida rakenteet manuaalisesti. Nyt aiheen laueus aiheuttaa sen, että sen puitteissa ei kyetä keskittymään ongelmien rajaamiseen puhumattakaan niiden perinpohjaisesta tarkastelu sekä mahdollisesta ratkaisemisesta.

Tietotekniikkaan ja informaatioteknologiaan liittyvien tutkimusten ongelmana on usein tekniikan nopea kehittyminen. Kärjistetysti voidaan sanoa, että se mikä tietyn ohjelman kohdalla ei ole mahdollista uusimassa kehitysversiona, saattaa olla mahdollista seuraavassa versiossa. Markkinoiden suurimpien ohjelmistojen kehitysversioita julkaistaan tyyppillisesti kerran tai kaksi vuodessa, ja esiin tulleita ongelmia korjaavia päivityksiä vieläkin useammin. Niin ollen ohjelmistojen kyvykkyyttä tarkasteleva tutkimus lähes poikkeuksetta laahaa perässä ohjelmien todellista kapasiteettia.

Tästä seuraa, että yhteentoimivuuden kykenevyyden kartoittaminen antaa vain lyhytkaista tietoa asian tilasta ja usein vain uusinta teknologiaa hyödyntävien sovellusten välisestä tiedonsiirrosta. Tämän perusteella tutkimuksen aiheesta ei voida tehdä ajallisesti kovin pitkälle tehtyjä johtopäätöksiä. Lähinnä tutkimus antaa nykytilanteen kuvan asiasta. Ratkaisuna ongelmaan voisi olla eräänlainen standarditesti, jonka tavoitteena olisi nopeasti kartoittaa eri ohjelmien tai näiden kehitysversioiden eroavaisuuksia sekä selvittää tiedonsiirron rajat.

Yritysten kannalta on taloudellisesti suotavaa, että jo ennen suunnitteluprojektin alkua tai hyvin varahaisessa vaiheessa kyetään tunnistamaan tieto- ja laskentamallin välisen tiedonsiirron ongelmakohdat. Siten on etukäteen tiedossa aikaa vievät kohdat joihin kohdentaa resursseja ja asettaa osaavimmat henkilöt ratkaisemaan ongelmakohtia. Standarditestin tuloksia voi ajatella eräänlaisena tietopankkina, jonka sisältöä voidaan hyödyntää suunnitteluprosessin ohjaamisessa. Lisäksi uuden ohjelmaversion kohdalla voidaan nopeasti saada selville, ratkaiseeko se aiemmin tiedonsiirtoprosessissa kohdattuja ongelmia.

Organisaation, joka haluaa saavuttaa kilpailullista etua tieto- ja laskentamallien tehokkaalla käytöllä, on kannattavaa tehdä entistä enemmän yhteistyötä ohjelmisto- ja teknologiavalmistajien kanssa. Ohjelmistovalmistajien kaupallinen etu on vastata organisaation tarpeisiin, minkä seurauksena vaatimalla parempaa IFC-pohjaista tiedonsiirtoa eri ohjelmien välillä on mahdollista saavuttaa sujuvampaa yhteentoimintaa tieto- ja laskentamallien välillä. Myös kehityspotentiaali on merkittävästi suurempi, kun organisaatio sisäistää ja ymmärtää kokonaisvaltaisesti tiedonsiirtoteknologian tuomat mahdollisuudet.

Lähdeluettelo

Coenders, J.L. 2007. Interfacing between parametric associative and structural software. Teoksessa: Xie, M. & Patnaikuni, I. (toim.). Proceedings of the 4th international conference on structural and construction engineering, ISEC-4 - Innovations in structural engineering and construction. Melbourne, Australia, 2007. Taylor & Francis. s. 63-68. ISBN 978-041-54575-5-2.

Coenders, J.L. 2010. Parametric and associative design as a strategy for conceptual design and delivery to BIM. Teoksessa: Lázaro, C & Domingo, A. IASS 2009 symposium report: 50th jubilee symposium of the international association for shell and spatial structures. Valencia, Espanja, 28.9.-2.10.2009. Editorial de la Universitat Politècnica de Valencia. Journal of the international association for shell and spatial structures. Vol. 50. S. 1112-1123. ISBN 978-848-36346-1-5.

Eastman, C., Teicholz, P. Sacks, R. & Liston, K. 2011. BIM Handbook: a guide to building information modeling for owners, managers, designers, engineers, and contractors. 2. painos. New Jersey, Yhdysvallat: John Wiley & Sons, Inc. 648 s. ISBN 978-047-05413-7-1.

Graphisoft. 2012. Interoperability with Structural Disciplines. Viitattu: 15.11.2017. Saatavissa: http://www.graphisoft.com/ftp/pdf/Structural_Interoperability.pdf

Haataja, J., Heikonen, J., Leino, Y., Rahola, J., Ruokolainen, J. & Savolainen, V. 2002. Numeeriset menetelmät käytännössä. Helsinki: CSC – Tieteellinen laskenta Oy. 415 s. ISBN 952-9821-81-6.

Hejnfelt, T. & Øksengaard, R. 2007. The use of 3D and BIM technology for structural analysis and design. Master's Thesis. Technical University of Denmark, Department of Civil Engineering. Kongens Lyngby. 131 s.

Hellsten, J. 2010. Todellinen tietomallin käyttö on vielä kaukana. Rakennuslehti.

Hietanen, J. 2005. Tietomallit ja rakennusten suunnittelu: filosofinen selvitys tieto- ja viestintätekniikan mahdollisuuksista. Helsinki: Rakennustieto. 95 s. ISBN 951-682-783-7.

Hämäläinen, J. & Järvinen, J. 2006. Elementtimenetelmä virtauslaskennassa. 2. painos. CSC – Tieteellinen laskenta Oy. ISBN 952-55201-9-6. Saatavissa: <http://www.csc.fi/csc/julkaisut/opaat/pdfs/elementtimenetelma>

Howell, I. & Batcheler, B. 2005. Building Information Modeling Two Years Later – Huge Potential, Some Success and Several Limitations. Viitattu 27.11.2017. Saatavissa: http://www.laiserin.com/features/bim/newforma_bim.pdf.

Ibrahim, M., Krawczyk, R. & Schipporeit, G. 2004. Two approaches to BIM: a comparative study. Yhdysvallat, Illinois Institute of Technology, College of Architecture.

Isikdag, U., Aouad, G., Underwood, J. & Wu, S. 2007. Building information models: a review on storage and exchange mechanisms. Yhdistynyt kuningaskunta, University of Salford, School of the Built Environment.

Jeong, Y.S., Eastman, C. Sacks, R. & Kaner, I. 2009. Benchmark tests for BIM data exchanges of precast concrete. *Automation in construction*. Vol. 18:4. s. 469–484. ISSN 0926-5805.

Kiviniemi, A, Karlshøj, J, Tarandi, V, Bell, H & Karud, O. 2008. Review of the Development and Implementation of IFC compatible BIM.

Koivu, T. 2002. Kiinteistö- ja rakennusalan tuotemallien ja yhteensopivuuden tulevaisuus – Vaihtoehtoisia skenaarioita ja teknologiapolkuja. Espoo: Valtion teknillinen tutkimuskeskus (VTT), Rakennus- ja yhdyskuntatekniikka. 53 s. ISBN 951-38-6080-9.

Korpela, J. 2011. Tietomallintamisen hyödyt ja haasteet rakennushankkeen eri osapuolten näkökulmasta. Diplomityö. Aalto-yliopisto, Insinööritieteiden korkeakoulu, rakennustekniikan laitos, rakentamistalous. Espoo. 66 s.

Kortelainen, J. 2011. Semantic data model for multibody system modelling. Väitöskirja. VTT. Espoo. 119 s. ISBN 978-951-38774-3-9.

Kouhia, R. & Tuomala, M. 2009. Rakennetekniikan numeeriset menetelmät luentomoniste. Viitattu: 27.11.2017. Saatavissa:

http://webhotel2.tut.fi/mec_tme/personnel/kouhia/papers/lecture_notes/pruju.pdf.

Kurowski, P.M. 2004. Finite element analysis for design engineers. [verkkoaineisto]. Warrendale, Pensylvania, Yhdysvallat: Society of Automotive Engineers (SAE) International. 176 s. ISBN 978-076-80114-0-1. Saatavissa: http://www.knovel.com/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid=3411&VerticalID=0

Kymmell, W. 2008. Building information modeling: planning and managing construction projects with 4D CAD and simulations. New York, Yhdysvallat: McGraw-Hill Companies. 270 s. DOI: 10.1036/0071494537. ISBN 978-0-07-149453-3.

Lee, G., Sacks, R. & Eastman, C. 2005. Specifying parametric building object behavior (BOB) for a building information modeling system. *Automation in construction*. Vol. 15:6. s. 758-776. ISSN 0926-5805.

Lähteenmäki, M. 2009. Elementtimenetelmän perusteet. [Verkkodokumentti]. [Viitattu 5.11.2017]. Saatavissa: http://personal.inet.fi/koti/mlahten/arkistot/elpe_pdf/johdanto.pdf

Mäki, T., Paavola, S., Kerosuo, H. & Miettinen, R. 2012. Tietomallintamisen käytöt rakentamisessa. Konsepti 7(1–2). [Verkkolehti]. Viitattu 27.11.2017. Saatavissa: http://www.muutoslaboratorio.fi/files/4_Tietomallintamisen_kayttoonoton_ongelmat_rakennushankkeessa.pdf

Niggel, A., Rank, E., Mundani, R.P. & Bungartz, H.J. 2006. A framework for embedded structural simulation: benefits in building design. Teoksessa: Smith, I.F.C. (toim.). Joint international conference on computing and decision making in civil and building engineering. Montreal, Canada, 14.-16.6.2006. Springer Berlin Heidelberg. s. 1768–1777. ISBN 978-354-04624-6-0.

Ojala, J. 2013. Interoperability in computer aided design. Diplomityö. Aalto-yliopisto, insinööritieteiden korkeakoulu, koneenrakennustekniikan laitos. Espoo. 63 s.

Pazlar, T. & Turk, Ž. 2008. Interoperability in practice: geometric data exchange using the IFC standard. *Electronic journal of information technology in construction*. Vol. 13. s. 362–380. ISSN 1403-6835. Saatavissa: <http://www.itcon.org/2008/24>.

Penttilä, H., Nissinen, S. & Niemioja, S. 2006. Tuotemallintaminen rakennushankkeessa: yleiset periaatteet. Tampere: Rakennusteollisuus RT ja Rakennustietosäätiö RTS. 64 s. Pro IT -hanke. ISBN 978-951-68279-6-7.

Polak, T.A. & Pande, C. 1999. Engineering measurements – methods and intrinsic errors. [verkkodokumentti]. Lontoo, Yhdistynyt kuningaskunta: Professional engineering publishing ltd. 195 s. ISBN 978-159-12464-5-9. Saatavissa: http://www.knovel.com.libproxy.aalto.fi/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid=922&VerticalID=0

Senescu R., Mole R. & Fresquez A. 2006. A case study in structural drafting, analysis and design using an integrated intelligent model. Joint International Conference on Computing and Decision Making in Civil and Building Engineering. Montreal, Kanada. 14.-16.6.2006.

Rammant, J.P. & Adriaenssens, G. 2008. Interoperability for BIM – a structural engineering viewpoint. Nemetschek Scia: Belgia, Herk-de-Stad. Viitattu: 27.11.2017. 12 s. Saatavissa: <http://nemetschek-scia.com/en/system/files/documents/WP-InteroperabilityBIM-300408-EN.pdf>

Romo, I. & Varis, M. 2004. Tuotemallinnus rakennesuunnittelussa – perusteet ja ohjeita I. 29.9.2004. Viitattu 27.11.2017. Julkaistu osana Pro IT projektia. Saatavissa: http://virtual.vtt.fi/virtual/proj6/proit/julkiset_tulokset/proit_rakennesuunnitteluohje_syysskuu2004.pdf

Romo, I., Karstila, K., Melvasalo, L., Niemioja, S. & Sulankivi, K. 2005. The use of product model data in building construction process, teoksessa: A.S. Kazi (toim.), ICT in construction and facilities management. Espoo: VTT. S. 93–104.

Sanguinetti, P., Abdelmohsen, S., Lee, J., Lee, J., Sheward, H. & Eastman, C. 2012. General system architecture for BIM: an integrated approach for design and analysis. *Advanced Engineering Informatics*. Vol. 26:2. S. 317–333.

Shephard, M.S., Beall, M.W., O'Bara, R.M. & Webster, B.E. 2004. Toward simulation-based design. *Finite Elements in Analysis and Design*. Vol. 40:2. S. 1575–1598. ISSN 0168-874X. Saatavissa: <http://www.sciencedirect.com/science/article/pii/S0168874X04000174>

Steel, J., Drogemuller, R. & Toth, B. 2010. Model interoperability in building information modelling. *Software & Systems Modeling*. Vol 11:1, s. 99-109. ISSN 1619-1366.

Succar, B. 2009. Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in construction*. Vol. 18:3. s. 357–375. ISSN 0926-5805.

Sulankivi, K. 2004. Kokemuksia tuotemallin ja 4D:n hyödyntämisestä pilottihankkeissa. Tampere. VTT Rakennus- ja yhdyskuntatekniikka. 54 s. Saatavissa: http://virtual.vtt.fi/proit/julkiset_tulokset/proit_pilottiraportti.pdf.

TSK. 2012. Tietotekniikan termitalkoot. Sanastokeskus TSK ry. Viitattu 27.11.2017. Saatavissa: <http://www.tsk.fi/tsk/termitalkoot>.

Valjus, J. 2007. Tuotemallintaminen rakennesuunnittelussa. Helsinki: Rakennustieto.

Wang, W.G., Tolk, A. & Wang, W.P. 2009. The levels of conceptual interoperability model: applying systems engineering principles to modeling and simulation. Proceedings of the Spring Simulation Multiconference. Society for Modeling & Simulation International (SCS), San Diego, CA, Yhdysvallat. Saatavissa: <http://arxiv.org/ftp/arxiv/papers/0908/0908.0191.pdf>